

World Scientific Monograph Series
in Mathematics – Vol. 4

Dynamics and Mission Design Near Libration Points

Vol. III Advanced Methods
for Collinear Points

G. Gómez

À. Jorba

J. Masdemont

C. Simó

World Scientific



Dynamics and Mission Design Near Libration Points

**Vol. III Advanced Methods
for Collinear Points**

World Scientific Monograph Series in Mathematics

Eds. Ron Donagi (*University of Pennsylvania*), Rafael de la Llave (*University of Texas*) and Mikhail Shubin (*Northeastern University*)

Published

Vol. 1: Almgren's Big Regularity Paper : Q-Valued Functions Minimizing Dirichlet's Integral and the Regularity of Area-Minimizing Rectifiable Currents up to Codimension 2

Eds. V. Scheffer and J. E. Taylor

Vol. 3: Dynamics and Mission Design Near Libration Points
Vol. II Fundamentals: The Case of Triangular Libration Points
by G. Gómez, J. Llibre, R. Martínez and C. Simó

Vol. 4: Dynamics and Mission Design Near Libration Points
Vol. III Advanced Methods for Collinear Points
by G. Gómez, À. Jorba, J. Masdemont and C. Simó

Vol. 6: Hamiltonian Systems and Celestial Mechanics
Eds. J. Delgado, E. A. Lacomba, E. Pérez-Chavela and J. Llibre

Forthcoming

Vol. 5: Dynamics and Mission Design Near Libration Points
Vol. IV Advanced Methods for Triangular Points
by G. Gómez, À. Jorba, J. Masdemont and C. Simó

World Scientific Monograph Series
in Mathematics – Vol. 4

Dynamics and Mission Design Near Libration Points

Vol. III Advanced Methods
for Collinear Points

G. Gómez, À. Jorba & C. Simó

Universitat de Barcelona, Spain

J. Masdemont

Universitat Politècnica de Catalunya, Spain



World Scientific

Singapore • New Jersey • London • Hong Kong



Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 912805

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

DYNAMICS AND MISSION DESIGN NEAR LIBRATION POINTS VOL. 3
Advanced Methods for Collinear Points

Copyright © 2001 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the Publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

ISBN 981-02-4211-5

Printed in Singapore by Uto-Print

Preface

In this work we study several problems related to the mission analysis of future planned or possible spacecraft missions. Most of the work deals with orbits near the collinear libration points, except the first chapter, where a continuation of the study in [3] near equilateral points is done. The work is divided in four chapters, each one dealing with one of the following topics:

- a) The first chapter is devoted to the computation of quasi-periodic solutions for the motion of a spacecraft near the equilateral points of the Earth–Moon system. The motion of the bodies of the Solar System is assumed to be as close as possible to the real one. To this end the ephemeris for the actual computations have been taken from the JPL tapes.
As the problem of computing quasi-periodic solutions is numerically badly defined, the following methodology has been used:
 - a.1) First we develop an analytic model for the equations of motion near the equilateral points. This is done by a Fourier analysis of the full equations of motion where the ephemeris of the solar system bodies (in that case Earth, Moon and Sun are the relevant ones) are given by available analytic expressions.
 - a.2) Then we look for an analytic quasi-periodic solution with the basic frequencies present in the equations (that is: the mean elongation of the Sun, the mean motions of the Earth and the Moon and the mean motions of the lunar perigee and node). This is done by using a suitable algebraic manipulator. Furthermore, we require that if in the analytic model we neglect the terms which are not in the RTBP model then we should find the equilateral relative equilibrium solution.
 - a.3) When the analytic approximation is available, we use a parallel shooting algorithm to obtain a numerical solution of the full equations of motion using ephemeris given by the JPL model. The end points of a given time interval are set equal to the ones found in the analytic solution. When the iterative process is finished we have a true orbit for

the spacecraft close to the analytic quasi-periodic solution. As a complementary result the stability properties for this orbit are obtained.

- b) The second chapter gives a complete description of the orbits near the collinear point, L_1 , between the Earth and the Sun in the RTBP model. It is known that the vicinity of L_1 is an ideal place for the location of a solar observatory. This was the task of the ISEE-C spacecraft and will be the one of the SOHO one in the next future. Both have a halo orbit as nominal orbit. Halo orbits for the RTBP have been already studied. Furthermore, the perturbations produced by the noncircular motion of Earth and Sun, the Moon and, to a lesser extend, the planets, are moderate. Hence one can use a combination of algebraic manipulation and parallel shooting, as described in Chapter 1, to obtain a numerical quasi-periodic solution using the JPL Ephemeris. Going back to the RTBP, which is the essential part of the problem, the following questions are addressed:
 - b.1) The halo periodic orbits (in the synodic reference system) are not the only interesting orbits near L_1 . There are also the quasi-periodic (eventually periodic) Lissajous orbits, lying on invariant tori. The halo orbits have the drawback of requiring a minimum amplitude in the planar direction orthogonal to the Earth-Sun line. This minimum amplitude is quite large. Other orbits confined to a lesser angular distance from the Sun (as seen from the Earth) are the Lissajous orbits. However they cross in front of the solar disc during their evolution. To assess the possibilities and requirements if Lissajous orbits are used, we have computed this two-dimensional family of orbits analytically at a very high order by using a suitable algebraic manipulator. The possible orbits and the advantages/disadvantages are discussed.
 - b.2) All the orbits in the vicinity of L_1 are highly unstable. This is the reason why one should use a careful station keeping to be studied in Chapter 3. This prevents also to obtain by direct numerical simulation a full description of the vicinity of L_1 . Furthermore, due to the 1 to 1 resonance giving rise to the halo orbits, the Lissajous orbits have been obtained in b.1) in a domain not reaching the halo orbits. To have a full picture of the possible motions we have to skip the unstable part of the motion. This is done by using the so called reduction to the central manifold (again via symbolic manipulation). It results a Hamiltonian system with two degrees of freedom which has been then numerically integrated and fully described. Beyond the Lissajous orbits there are also quasi-periodic orbits around the halo orbits, which can also be useful as nominal orbits.
- c) In Chapter 3 methods to compute the nominal orbit and to design and test the control strategy for the quasi-periodic halo orbits are developed.

Now we return to the full solar system model given by the JPL ephemeris including the solar radiation pressure. The nominal orbit considered is a quasi-periodic perturbation of a periodic orbit of the RTBP. As it is unstable it requires a station keeping which takes into account the behavior near the nominal orbit. In the full process we distinguish several steps:

- c.1) To obtain the one-parameter family of halo orbits for the RTBP. This was already done in a previous work for the ESA [2] by using algebraic manipulation. The variational equations along the halo orbit were also integrated analytically by suitable series expansions.
- c.2) The halo orbit has to be modified to include all the quasi-periodic perturbations to be added to the RTBP. This can be done, as in a), by obtaining analytically some terms and then by improving the results using a parallel shooting with JPL Ephemeris, or performing directly the parallel shooting (with the eventual use of a continuation parameter to ensure convergence of the method). The first approach has been taken, using some analytic quasi-periodic terms as found in [2]. The numerical refinement has been improved to cover long time periods. The orbit obtained depends on a vertical amplitude and an initial epoch.
- c.3) The escape from the vicinity of the nominal halo orbit takes place along the unstable direction. When the tracking allows us to assure that such an escaping is taking place a manoeuvre should be done. The direction and the amount of this manoeuvre are easily computed if the solution of the variational equations along the nominal orbit are available. We follow again the methodology of [2]. The (x, y) manoeuvres are the ones with a larger gain and we have restricted to this case.
- c.4) To test the strategy of the station keeping for a nominal halo orbit, several runs have been done. In those runs we have taken different magnitudes for the distribution of errors in tracking, execution of manoeuvres and effect of solar radiation pressure. We have also checked the effect of the changes in the nominal orbit and in the criteria to decide when to do a manoeuvre (or possible delays in the execution). The results are excellent (roughly only a 2.5% of the real amount of Δv used for the ISEE-C is enough) and very robust. Furthermore the different runs allow to measure the effect of each one of the sources of error.
- d) In Chapter 4 the transfer from the Earth to a halo orbit is studied. This problem can be seen in the general case as the one of reaching a nominal orbit by suitable manoeuvres. Then an optimization process is started (from an initial guess or approximated solution) to minimize the fuel consumption

under some scientific or technical constraints. Typically one should do a big manoeuvre to inject the spacecraft in a transfer orbit, a mid course correction manoeuvre to correct for the errors of the previous one, and a final manoeuvre to insert into the nominal orbit. Our approach has been different. The target orbit has a stable manifold. It turns out that this manifold comes close to the Earth (without further passages near the Earth) at some times and for suitable initial epochs of the halo orbit. The work is divided as follows:

- d.1) Given a vertical amplitude and an initial epoch the halo orbit is obtained as in c). Then the local stable manifold is approximated (at a given time) by integration of the variational equations obtaining the stable direction.
- d.2) The manifold is globalized by backwards numerical integration (using the JPL ephemeris, of course). The orbits in the stable manifold leading to close approaches to the Earth are stored.
- d.3) Additional computations of orbits in the stable manifold are done in order to have a finer mesh. Then the orbits suitable for the transfer are selected. To this goal one should skip orbits passing too close to the Moon, or reaching the vicinity of the Earth with a too low or a too high perigee distance or with a nonallowable equatorial inclination. In this way one has the selection of the transfer possibilities and the amount of the manoeuvre to inject on the stable manifold. The mid course corrections to prevent from injection errors are unavoidable, but with this approach one can skip the injection manoeuvre in the nominal orbit.

Additional information is given in the Appendices A to D including comments on the implementation of the JPL Ephemeris (Appendix A) and on the reference systems and equations of motion (Appendix B). They also give the derivation of the analytic model used in a) (Appendix C) and an study of the transfer between orbits of the halo family for the RTBP which allows to change easily the nominal orbit in situ (Appendix D).

We should like to conclude this Introduction by making some comments on the methodology. The general point of view is the Dynamical Systems approach, mainly Hamiltonian Systems with time-dependent perturbations. The semianalytic approach (i.e. the use of algebraic manipulators with numerical coefficients) has been carried out as far as possible to obtain information not only about one orbit but a big amount of them. This has been combined with refined numerical methods to get real solutions to real problems. Furthermore the methods are presented in such a way that its use in a big variety of similar problems becomes easy.

The contents of this book is the final report of the study contract that was done for the European Space Agency in 1991. This report is reproduced textually with

minor modifications: the detected typing or obvious mistakes have been corrected, some tables have been shortened and references, which appeared as preprints in the report, have been updated. The layout of the (scanned) figures has changed slightly, to accommodate to latex requirements.

The last page of this preface reproduces the cover page of the report for the European Space Agency showing, in particular, the original title of the study.

For the ESA's study we also produced software that is not included here, although all its main modules are described in detail in the text.

Updates on the state of the art, both concerning theoretical and practical studies, can be found at the end of Volume IV of this collection of works on *Dynamics and Mission Design Near Libration Points*.

This page is intentionally left blank

Contents

Preface	v
Chapter 1 Quasi-periodic Solutions Near the Equilateral Points of the Earth–Moon System	1
1.1 Introduction	1
1.1.1 Expansion of the Equations	2
1.2 Idea of the Resolution Method	4
1.3 The Algebraic Manipulator	4
1.3.1 Storing Fourier Series	5
1.3.2 Basic Subroutines	6
1.3.3 Making Operations	7
1.4 The Newton Method	9
1.4.1 Some Remarks about the Jacobian Matrix	9
1.4.2 Adding Terms to the Newton Process	10
1.4.3 High Level Routines	10
1.5 The Program	13
1.5.1 Test of the Program	15
1.6 Results of the Algebraic Manipulator	15
1.7 Numerical Refinement	21
1.7.1 Final Results	22
1.8 The Neighbourhood of the Computed Nearly Quasi-periodic Solution . .	25
1.9 Problems and Extensions	31
Chapter 2 Global Description of the Orbits Near the L_1 Point of the Earth–Sun System in the RTBP	33
2.1 Introduction	33
2.2 The Equations of Motion	33
2.3 Formal Series Solutions	36
2.3.1 A Program to Obtain the Formal Series Solutions	39
2.3.2 Results and Formal Tests	41

2.3.3	Numerical Tests	47
2.4	On the Convergence of the Series	51
2.5	Towards a Description of the Neighbourhood of L_1	53
2.6	Discussion on the Use of Lissajous Orbits	54
2.7	Effective Reduction to the Central Manifold	56
2.7.1	The Method	56
2.7.2	The Program and the Results	60
2.7.3	Numerical Simulations and Tests	63
2.8	Conclusions	66
Chapter 3	Quasi-periodic Halo Orbits	69
3.1	Numerical Refinement	69
3.1.1	Computation of the Initial Values	70
3.1.2	The Differential Matrix $DF(Q)$	72
3.1.3	Computing the Poincaré Map and its Differential	72
3.2	Main Program and Basic Routines	77
3.2.1	Numerical Results	78
3.3	The Equations of Motion for the Simulations of the Control	83
3.4	The Effect of Errors	83
3.5	When a Control is Applied	84
3.6	Magnitudes Related to the Control	85
3.7	Description of the Program	86
3.8	Numerical Results	88
3.8.1	Discussion on the Effect of the Different Errors and Magnitudes	91
3.8.2	Conclusions	91
Chapter 4	Transfer From the Earth to a Halo Orbit	93
4.1	Introduction	93
4.2	Local Approximation of the Stable Manifold	94
4.2.1	Main Program and Basic Routines	96
4.2.2	Output and Sample of Results	97
4.3	Globalization of the Manifold	100
4.3.1	Main Program and Basic Routines	101
4.3.2	Output and Sample of Results	101
4.4	Selecting Passages Near the Earth	104
4.4.1	Main Program and Basic Routines	105
4.4.2	Output and Sample of Results	106
4.5	Ranges in the Manifold Suited for the Transfer	108
4.5.1	Main Programs and Basic Routines	108
4.5.2	Output and Sample of Results	109
4.6	Characteristics of the Orbits Near the Earth	114
4.6.1	Main Program and Basic Routines	114
4.6.2	Output and Sample of Results	114

4.6.3	Some Numerical Explorations	117
4.7	Conclusions	126
Appendix A The JPL Model		127
A.1	Implementation and Use	127
A.2	Basic Subroutines	130
Appendix B Reference Systems and Equations of Motion		133
B.1	The Restricted Three Body Problem	133
B.2	The Real Problem	137
B.2.1	Reference Systems	137
B.2.2	Equations of Motion	139
Appendix C The Model Equations Near the Equilateral Points in the Earth–Moon System		147
C.1	Introduction	147
C.2	Systems of Reference	147
C.3	The Lagrangian	149
C.4	The Hamiltonian and the Related Expansions	150
C.5	Some Useful Expansions	151
C.6	Fourier Analysis: The Relevant Frequencies and the Related Coefficients.	154
C.7	Simplified Normalized Equations	164
C.8	Numerical Tests	170
Appendix D Transfer Between Halo Orbits of the RTBP		177
D.1	Introduction	177
D.2	Strategy of the Method	177
Bibliography		187

This page is intentionally left blank

Chapter 1

Quasi-periodic Solutions Near the Equilateral Points of the Earth–Moon System

1.1 Introduction

The main purpose of this chapter is to find quasi-periodic solutions near the equilateral libration points of the Earth–Moon system.

The equations of motion near these points can be written as (see [3])

$$\ddot{x} = P(7) \left[-\frac{x - x_E}{r_{PE}^3} (1 - \mu_M) - \frac{x + x_E}{r_{PM}^3} \mu_M - x_E (1 - 2\mu_M) \right] + P(1) + P(2)x + P(3)y + P(4)z + P(5)\dot{x} + P(6)\dot{y}, \quad (1.1)$$

$$\ddot{y} = P(7) \left[-\frac{y - y_E}{r_{PE}^3} (1 - \mu_M) - \frac{y - y_E}{r_{PM}^3} \mu_M - y_E \right] + P(8) + P(9)x + P(10)y + P(11)z + P(12)\dot{x} + P(13)\dot{y} + P(14)\dot{z}, \quad (1.2)$$

$$\ddot{z} = P(7) \left[-\frac{z}{r_{PE}^3} (1 - \mu_M) - \frac{z}{r_{PM}^3} \mu_M \right] + P(15) + P(16)x + P(17)y + P(18)z + P(19)\dot{y} + P(20)\dot{z}, \quad (1.3)$$

where r_{PE} , r_{PM} denote the distances from the particle to the Earth and Moon, respectively, given by $r_{PE}^2 = (x - x_E)^2 + (y - y_E)^2 + z^2$, $r_{PM}^2 = (x + x_E)^2 + (y - y_E)^2 + z^2$. We recall that $x_E = -1/2$, $y_E = -\sqrt{3}/2$ for L_4 and $x_E = -1/2$, $y_E = \sqrt{3}/2$ for L_5 .

The functions $P(i)$ that appear in the equations are of the form:

$$P(i) = A_{i,0} + \sum_{j=1}^m A_{i,j} \cos \theta_j + \sum_{j=1}^m B_{i,j} \sin \theta_j,$$

with $\theta_j = \nu_j t + \varphi_j$, where t denotes the normalized time. Values for all of the coefficients, frequencies and phases can be found in Appendix C.

An expansion of the nonlinear terms of the equations of motion, in power series of the coordinates (x, y, z) , can be done easily using Legendre polynomials as it is described in Section 1.1.1.

Another possibility is to produce those terms by means of a recurrence based on the one of the Legendre polynomials.

1.1.1 Expansion of the Equations

We consider the potential $V = \frac{1-\mu}{r_{PE}} + \frac{\mu}{r_{PM}}$. Let ψ be the angle between the vectors $(x_E, y_E, 0)$ and (x, y, z) and ρ^2 be $x^2 + y^2 + z^2$. Then

$$\frac{1}{r_{PE}} = \frac{1}{\sqrt{1 - 2\rho \cos \psi + \rho^2}} = \sum_{n=0}^{\infty} \rho^n P_n(\cos \psi), \quad (1.4)$$

where P_n is the Legendre polynomial of degree n :

$$P_n(\omega) = \frac{1}{2^n n!} \frac{d^n}{d\omega^n} (\omega^2 - 1)^n = \sum_{k=0}^{\left[\frac{n}{2}\right]} \alpha_{n,k} \omega^{n-2k}.$$

Here $\left[\frac{n}{2}\right]$ denotes the integer part of $\frac{n}{2}$ and

$$\alpha_{n,k} = \frac{(-1)^k}{2^n n!} \binom{n}{k} \frac{(2n-2k)!}{(n-2k)!}.$$

Substituting these expressions in (1.4) one obtains

$$\begin{aligned} \frac{1}{r_{PE}} &= \sum_{n=0}^{\infty} \rho^n \sum_{k=0}^{\left[\frac{n}{2}\right]} \alpha_{n,k} \cos^{n-2k} \psi \\ &= \sum_{n=0}^{\infty} \sum_{k=0}^{\left[\frac{n}{2}\right]} \alpha_{n,k} (xx_E + yy_E)^{n-2k} (x^2 + y^2 + z^2)^k. \end{aligned} \quad (1.5)$$

Finally, in order to obtain the expansion for L_4 (the L_5 case can be done with an identical process) we put $x_E = -\frac{1}{2}$, $y_E = -\frac{\sqrt{3}}{2}$ and (1.5) becomes

$$\frac{1}{r_{PE}} = \sum_{n=0}^{\infty} \sum_{k=0}^{\left[\frac{n}{2}\right]} \beta_{n,k} (x + \sqrt{3}y)^{n-2k} (x^2 + y^2 + z^2)^k, \quad (1.6)$$

where

$$\beta_{n,k} = \frac{(-1)^{n-k} (2n-2k)!}{4^{n-k} (n-k)! k! (n-2k)!}.$$

To perform the expansion (1.6) by algebraic manipulation it is better to change the

order of the sums:

$$\frac{1}{r_{PE}} = \sum_{k=0}^{\left[\frac{N}{2}\right]} (x^2 + y^2 + z^2)^k \sum_{n=2k}^N \beta_{n,k} (x + \sqrt{3}y)^{n-2k},$$

where N denotes the expansion order wanted. With a similar computation it can be found that:

$$\frac{1}{r_{PM}} = \sum_{k=0}^{\left[\frac{N}{2}\right]} (x^2 + y^2 + z^2)^k \sum_{n=2k}^N \beta_{n,k} (-x + \sqrt{3}y)^{n-2k}$$

and, finally,

$$\begin{aligned} V &= \frac{1-\mu}{r_{PE}} + \frac{\mu}{r_{PM}} = \sum_{k=0}^{\left[\frac{N}{2}\right]} (x^2 + y^2 + z^2)^k \\ &\times \sum_{n=2k}^N \beta_{n,k} \left[(1-\mu)(x + \sqrt{3}y)^{n-2k} + \mu(-x + \sqrt{3}y)^{n-2k} \right]. \end{aligned} \quad (1.7)$$

Now, using a program which takes advantage of the particularities of the latter expression, the expansions of $\frac{\partial V}{\partial x}$, $\frac{\partial V}{\partial y}$ and $\frac{\partial V}{\partial z}$ are obtained. This program has been written to give the final solution up to order nine. This is enough for practical purposes, because the contribution of higher order terms is much less than the threshold used to select the coefficients $A_{i,j}$ and $B_{i,j}$. Its routines are:

Subroutine DERIV Given a polynomial of three variables x , y and z , this routine returns three polynomials which are its derivatives with respect to x , y and z . This is used at the end of the computation of (1.7) to obtain the expansion wanted.

Subroutine PRODE This routine performs the product of two polynomials of three variables. It is used to multiply the term $(x^2 + y^2 + z^2)^k$ by the result of the inner summation of (1.7).

Subroutine XYZK Given a polynomial S , this routine returns $S \times (x^2 + y^2 + z^2)$. It is used to obtain the different powers of $(x^2 + y^2 + z^2)$ starting from $S = 1$.

Subroutine PART2V This routine computes the result of the inner summation of (1.7), taking advantage of the cancellations that occur in this formula

Subroutine INIT This routine fills a real array with the powers of $\sqrt{3}$ (they will be used inside PART2V) and a couple of integer arrays containing information about the way to store polynomials (in fact this is not necessary, but it allows to increase the speed of the program).

Main Program This routine initializes the parameters used by the program (x_E , y_E and μ) and it uses the formula (1.7) to compute V . Then the final result is obtained (by derivation) and the result is written in the files EXPAN.1

(expansion of the first equation), EXPAN.2 (second one) and EXPAN.3 (third one).

1.2 Idea of the Resolution Method

The equations of motion can be written as

$$\begin{aligned}\ddot{x} &= f_1(t, x, y, z, \dot{x}, \dot{y}, \dot{z}), \\ \ddot{y} &= f_2(t, x, y, z, \dot{x}, \dot{y}, \dot{z}), \\ \ddot{z} &= f_3(t, x, y, z, \dot{x}, \dot{y}, \dot{z}).\end{aligned}$$

Define $G = (G_1, G_2, G_3)$, where $G_1 = f_1 - \ddot{x}$, $G_2 = f_2 - \ddot{y}$, and $G_3 = f_3 - \ddot{z}$. We are looking for quasi-periodic solutions of the form

$$\begin{aligned}x &= \sum x_k \exp((k, \omega)t\sqrt{-1}), \\ y &= \sum y_k \exp((k, \omega)t\sqrt{-1}), \\ z &= \sum z_k \exp((k, \omega)t\sqrt{-1}),\end{aligned}$$

where $\omega = (\omega_1, \omega_2, \dots, \omega_r)^t$ is a known set of basic frequencies, the ones that appear as basic frequencies, in the developments of the functions $P(i)$, and (k, ω) denotes the inner product of $k = (k_1, k_2, \dots, k_r)^t$ and ω . We consider x , y and z truncated up to some order. After substitution, G may be considered as a function of x_k , y_k and z_k , which means that looking for quasi-periodic solutions of this problem is more or less equivalent (this depends on the order of approximation taken for the trigonometric expansions of x , y , and z) to look for a zero of the function $G : \mathbf{R}^m \rightarrow \mathbf{R}^m$, where m denotes the total number of coefficients. To solve this equation we use a Newton method. In order to calculate the Jacobian matrix, the chain rule is used in the following way: $\frac{\partial G}{\partial x_k} = \frac{\partial G}{\partial x} \cdot \frac{\partial x}{\partial x_k}$, and so on.

1.3 The Algebraic Manipulator

Now we are going to describe the algebraic manipulator specially built (in FORTRAN 77 language) to solve the latter problem. Here, a trigonometric series is stored as a vector of real numbers (each one corresponding to a coefficient), a vector of integers (of the same dimension) and an integer containing this dimension. This integer is only used in some operations to check an “overflow trigonometric series” condition. An auxiliary table, with the linear combinations of the basic frequencies, will also be used by the manipulator.

1.3.1 Storing Fourier Series

We need to store trigonometric series and handle them in a fast way. Moreover, we want the size of the vectors that contain them to be as short as possible. In order to get the latter condition, we have decided to store only the non-zero coefficients, positioned in a contiguous way (as a stack). This means that we must have a way to recognize those coefficients, because their position inside the vector does not allow to identify them.

For this reason we have taken an auxiliary integer table of frequencies which contains, for each column, different integer values. For example, if a column contains the integers (k_1, \dots, k_r) it means that it refers to the frequency $k_1\omega_1 + \dots + k_r\omega_r$, where $(\omega_1, \dots, \omega_r)$ is the vector of basic frequencies (obviously, the number of rows must be at least the number of frequencies). This table will be used in two different ways:

- (1) Given a position, to find the corresponding coefficients of the frequencies.
- (2) Given an integer vector, to search for its position inside the table.

The first use is easy and fast to do, but not the second one. At this point, it would be useful that elements of the table were given in some order, because we could use a binary search. One could think that it would be enough to fill the table at the beginning of the program using the selected order, but this does not work very well, due to the fact that when we start the program we do not know yet which frequencies will appear in the intermediate computations. We need to add frequencies to the table during the process, and we need to restart the order in a quick way. For this reason we have added to the table an integer vector which refers to the table in an ordered way and, when a new frequency is added (at the end of the table), the program only needs to modify this vector to get the table in order again. The order selected has been lexicographic:

$$(k_1^1, \dots, k_r^1) < (k_1^2, \dots, k_r^2) \Leftrightarrow \begin{cases} k_1^1 < k_1^2 & \text{if } k_1^1 \neq k_1^2 \\ (k_2^1, \dots, k_r^1) < (k_2^2, \dots, k_r^2) & \text{if } k_1^1 = k_1^2. \end{cases}$$

Now, let us see how we can use this. We suppose that the real vector of coefficients is called \mathbf{x} and we define an integer vector called \mathbf{nx} . The first component of \mathbf{x} is always used to store the independent term (the average). The first component of \mathbf{nx} is used to store the number of non-zero coefficients of the series (the last meaningful component of vector \mathbf{x}). Now, let j be an integer less than or equal to $\mathbf{nx}(1)$ and we consider the coefficient $\mathbf{x}(j)$. If $\mathbf{nx}(j)$ is even, $\mathbf{x}(j)$ corresponds to a cosine, otherwise it corresponds to a sine. In the column $\mathbf{m}=\mathbf{nx}(j)/2$ (integer division) of the table described above we can find the frequencies corresponding to this coefficient.

Now we are going to see how all of these things have been programmed. In what refers to storing frequencies in the table, we restrict ourselves to the ones that have the first integer coefficient (k_1 using the above notation) greater than or equal to

zero. If we have a frequency with $k_1 < 0$, we change its sign and, in the case that it corresponds to a sine, the sign of the coefficient is also changed. If $k_1 = 0$ we impose that the latter condition holds for k_2 . If $k_2 = 0$ we impose it to k_3 , and so on.

All of these arrays have been included inside a common called `taules`:

```
common /taules/ taufre,ntau,ordtau,busc
```

`taufre` has been declared as `integer*2` and `ntau`, `ordtau` and `busc` as `integer*4`. `taufre(r, n)` contains the table of frequencies (n is the maximum number of them and r the number of basic frequencies), `ntau` is the actual number of frequencies in the table, `ordtau(n)` is the vector that holds the order of `taufre` and `busc($2, m$)` is an array used to get a quicker binary search. Let us see how it works. Given a frequency (k_1, \dots, k_r) , $k_1 \geq 0$ the numbers `busc($1, k_1 + 1$)` and `busc($2, k_1 + 1$)` refer to the first and last place (of `ordtau`) where this k_1 appears, and then the search field is shorter. If there do not exist any frequencies with this k_1 the corresponding values of `busc` are -1 .

Finally, we are going to comment on some points about another system of storing Fourier series used in some intermediate computations. As we will see later, in many operations the manipulator uses a real array as a workspace. This vector has twice more components than the table of frequencies and it is used to hold series in such a way that it allows to make sums faster. This is due to the fact that the position of a coefficient in this vector plays the same role that the index vector seen before. For example, with the same notation as before, the coefficient $x(j)$ is put in the component `nx(j)` (this is only true for $j > 1$, $x(1)$ is put in the first component of the working vector). Note that if we have two vectors of this kind, the sum of them can be done by adding, component by component, the two vectors. The disadvantage of having many vectors like this is the waste of memory that they require, because many of these components may be zero. For this reason we prefer to keep the Fourier series in the packed form seen before. We have only one working vector and we pass it to routines that make use of it.

1.3.2 Basic Subroutines

The following routines have been built with the aim of having an easy-to-use system to handle the tables of frequencies.

Subroutine `FREQ` The first parameter is an `integer*4` variable coming from a Fourier series (for example, using the above notation, some value `nx(k)`) that contains information relative to a coefficient. Unchanged on exit. The second parameter is an `integer*4` array that will contain the frequencies corresponding to `n`, and the third parameter will contain 0 if it refers to a cosine or 1 if it refers to a sine.

Subroutine `POSI` This routine is inverse of the former one: it returns an integer

which contains information about the frequency (following with the example, information to be stored in the vector \mathbf{nx}). The first parameter is an `integer*4` vector of frequencies and the second one is an integer containing 0 or 1. These parameters are unchanged on exit. The routine returns an integer value n to be stored in an integer vector corresponding to a Fourier series. The routine performs a binary search of \mathbf{fv} in the table of frequencies and, assuming that \mathbf{fv} is found in the place j , makes $n=2*j+i$. If \mathbf{fv} is not found (i. e., it is not in the table) the routine calls the subroutine ACTUAL.

Subroutine ACTUAL This routine adds a frequency to the table. The first parameter is the vector containing the frequency to be added and the second one (`llloc`) is the place (of `ordtau`) where this frequency must be (this value has been found during the binary search of the routine POSI). The elements which previously were stored in `ortau`, in places from `llloc` up to the end, are shifted one place to the right. Both parameters are unchanged on exit.

Subroutine EMPAQ As we have already seen, in many operations the manipulator uses a real array as a workspace. This routine packs a trigonometric series contained in the working vector.

Subroutine NORMAL The first parameter is an integer vector (\mathbf{f}) containing a frequency, the second one is an integer saying how many frequencies we have. With this, the routine puts \mathbf{f} in normal form (the first component nonequal to zero must be positive) and returns `is=-1` if it has changed the sign of \mathbf{f} , or `is=1` otherwise. If all of the components of \mathbf{f} are equal to 0 it returns `is=0`. This routine is called during some operations (for example, a product of series).

Subroutine WFREC This routine returns the numerical value of the frequency corresponding to a place (given by means of a parameter) of the working vector.

Subroutine INITAU This routine initializes the tables used by the manipulator. It must be called at the beginning of the program, before making any other call to a routine of the manipulator. It has no parameters.

Subroutine AVALUA This routine computes the numerical value of a trigonometric series at a given time value t .

Subroutine LLEGEIX Given a channel (already open), this routine reads a Fourier series through this channel and returns it in packed form.

Subroutine ESCRIU Given a channel (already open), this routine writes a Fourier series given in packed form.

1.3.3 Making Operations

Now we are going to describe the main operations performed by the manipulator, which include sums, multiplications of Fourier series and multiplications of Fourier series by real numbers, differentiation and some special operations that we shall describe later. To keep only the meaningful terms of the series, we have included

a threshold that allows to drop terms less (in absolute value) than this threshold, which is in a common named `control`.

Subroutine SUMA This routine performs the sum of two Fourier series. The method consists in expanding the first series inside the working array and then add the second series putting the results in the same working vector. Finally, the result is packed (this routine calls `SUMAF`).

Subroutine SUMAF Given a Fourier series in packed form and a Fourier series in expanded form, this routine adds them and puts the result in the same vector that contains the series in expanded form (the vector in expanded form is the working vector mentioned in Section 1.3.1). This routine has been built to increase the speed of operations of type $a = a + b$.

Subroutine PROD This routine obtains the product of two series. The way to do this is to fill a working vector with zeros and to call the routine `PRODF` (see below). This routine returns the result inside the working vector and finally it is packed.

Subroutine PRODF Given two Fourier series in packed form (we can call them a and b) and a Fourier series in expanded form called w , this routine performs the operation $w = w + a \times b$. To do this, each term of the first series is multiplied by each term of the second one. If the absolute value of the resultant coefficient is less than the threshold, it is dropped, otherwise the frequencies of the terms that are multiplying are computed. With this, the result is added to the corresponding position of the working array. As `SUMAF`, this routine has been built to increase the speed of some operations.

Subroutine PR1C This routine performs the product of a series containing only one term (whose coefficient is equal to one) by a Fourier series in packed form. The result is returned also in packed form. The unitary series is given by a number displaying the frequencies and if it is a sine or a cosine (this number is of the same kind as the components of the index vector (`nx`) seen before). The routine works with the same algorithm as `PROD`. This operation is needed a lot of times during the calculus of the Jacobian matrix. This routine can be avoided, but it is easy to write and allows to simplify the main algorithm (we do not need to fill a trigonometric array with an unitary series and to call `PROD`).

Subroutine PR1CF This routine makes the same operation as `PR1C`, but adding the result to the series contained in the working vector. This allows us to get more speed during some operations.

Subroutine PRESC This routine multiplies a real number by a trigonometric series. This is a very common operation in the main algorithm and we prefer to write it instead of using the routine `PROD`.

Subroutine DF This routine computes the derivative of a Fourier series as usual: for each term the integer vector k containing the frequency is obtained, and the coefficient is multiplied by (k, ω) (the vector ω is stored in a common

called `frequ`) and if it is a cosine, its sign is changed.

Subroutine MDF2 This routine performs the second derivative of a trigonometric series, and multiplies it by -1 . Note that it is faster to compute this directly than to call routine `DF` twice and then to call `PRESC`. The algorithm is of the same kind that in the latter routine.

Subroutine NORMA This routine returns the norm of a Fourier series. The norm that we have selected is the Euclidean norm of the vector containing its coefficients.

1.4 The Newton Method

Now, using the manipulator described above, it is possible to write a program that looks for a zero of the function G defined before, by means of a Newton method. For this purpose we need an initial condition close enough to the solution that we are looking for. Due to the fact that the perturbation is too big to take the zero solution as initial condition, we are going to use a Newton continuation method. For this reason we have added a continuation parameter (h) multiplying the nonconstant terms of the perturbations. Thus, when h is equal to 0 we have an autonomous differential system with an equilibrium point near the origin (the equilibrium point is not the origin itself because the constant terms of the perturbations are not multiplied by h). This point is close enough to the origin to be found by the Newton process starting from the zero solution. Then we increase h and this point becomes a small quasi-periodic orbit (see [6]) that can be found by the Newton method (otherwise the value of h must be reduced to an intermediate value). Now the value of h is increased again and the Newton process is started from the solution found for the last value of the parameter. If no problems appear (that is, bifurcations and/or turning points), this process can continue until h reaches the value 1. We shall come back to this point later.

1.4.1 Some Remarks about the Jacobian Matrix

Due to the big quantity of coefficients appearing in the Fourier series, it is impossible to include all of them in the Newton method. For this reason we have selected some of them. The selected elements are stored in integer vectors (we store the indices corresponding to them as the vector `nx` seen before) and put in a common called `frenew`. Because of this, we have two different values of the function G : the ordinary value and the one obtained taking into account only the terms included in the Newton method. Note that we are solving with respect to this last value.

1.4.2 Adding Terms to the Newton Process

The criterion to select the coefficients that will take part in the Newton process is based on the size of the amplitudes of the residual acceleration. At the beginning only the terms appearing in the initial condition are selected (if this is constant and equal to zero, only the independent terms take part in the process). When the reduced equation (that is, taking into account only the terms of the Newton process) is solved, the program checks the size of the amplitudes of the terms of the residual acceleration that are outside the Newton process. If it finds that some of them are larger than a given threshold, then they must be added to the Newton process, else this step of the continuation method is finished and we can start the next one.

We want to make a remark about adding terms to the Newton process. If we add too many terms to it, we can find out that the neighborhood of convergence is too small and we are out of it. In order to avoid this, we have a control on the maximum number of terms going in the Newton process, and if this number is reached, the threshold is increased. We have also implemented a similar procedure if the program is going to take a few new terms for the Newton method. Of course, in this case the corresponding threshold is reduced.

1.4.3 High Level Routines

We start writing a routine that evaluates the function G . This is not a problem, using the algebraic manipulator and a series expansion for the terms coming from the RTBP. We also have continuation parameters in the terms of each degree and an integer variable containing the actual degree of the power expansion. These parameters are in a common called `conti`. This allows to select the degree at which the solution is wanted.

Then, we need a routine that computes the Jacobian matrix of G . This is easy using the chain rule as mentioned in 1.2 and the polynomial expansion of the equations. Note that in this part we also have the continuation parameters for the expansion. Moreover, in order to increase the speed, we realize that some powers and products of series that we need in this part have already been done during the evaluation of G , so, we keep these results in memory.

Once the Jacobian matrix is filled up, we need to solve a linear system with a big matrix (note that this matrix is not sparse at all). This system involves only the terms selected before. For this purpose, we have written a routine adapted to a paged environment. The method used is based on the Gaussian elimination.

Due to the fact that we do not know the stepsize for the continuation parameter, we have implemented an automatic stepsize control. This works in the following way: when the Newton method does not find the zero in a few iterations (this number is supplied by a control input file), it assumes that the continuation parameter has been increased too much. Then the program recovers the solution correspond-

ing to the last value of the parameter (it has been stored previously by the program in a disk file), assigns to the parameter an intermediate value and starts the process again. Of course, if the difference between this intermediate value and the last successful one is less than a fixed threshold (given also by a control file) the program stops issuing the corresponding error message.

Subroutine INIPOT This routine computes the required powers $x^i y^j z^k$, where (x, y, z) is a vector of Fourier series, and stores the result (in the common /pow/) to be used by the routines that compute the function G and its Jacobian matrix. If n is the actual degree of the expansion we are working with, then the powers are computed and stored* up to order $n - 1$. This is due to the fact that the powers of order n are only needed once (during the computation of G), while the ones of order less than n are used many times (in the computations of G and its Jacobian matrix). The way to obtain the powers of x , y and z is the following: powers of orders less or equal than three are computed “explicitly” (using the algebraic manipulator, we have written directly the expressions for order two and, using these results, the ones of order three). The powers of degree greater or equal than four are obtained as a product of two powers previously computed. This routine calls POSAP, TREUP and SPLIT.

Subroutine POSAP This routine stores a Fourier series $a = x^i y^j z^k$ in the common /pow/. The parameters are the Fourier series a , ia , na (see Section 1.3.1) and i , j , k . The routine puts the Fourier series in the vectors `real*8 pot` and `integer*4 npot`. This common also has an `integer*4` value containing the first free component of the vectors, in order to add the Fourier series at the end of the series already contained in the vectors. Once the series is added, this value is updated. To recover these series we need to write down where they are. This is done by means of the routine LLP (see below): given the integer vector of exponents i , j and k , this routine returns an integer value l . Then, in the column l of the array `integer*4 inpo(2,161)` we store the first and the last component of `pot` and `npot` in which we can find that Fourier series.

Subroutine LLP Given the integer vector (i, j, k) , the routine returns the column of `inpo` in which we can find the first and the last components of the arrays `pot` and `npot` containing the value $x^i y^j z^k$. More concretely, if $n = i + j + k$, then the returned value is $((n+3)n+2)n-24)6+i(n+1)-((i-1)i)/2+j+1$.

Subroutine TREUP This routine extracts the Fourier series of $x^i y^j z^k$. In order to know where it is stored, a call to the routine LLP is done.

Subroutine TREUPP This routine works exactly as TREUP but returning the result multiplied by a given `real*8` variable. It is used by the routine MONOMI (see below).

*except those of degree one

Subroutine SPLIT Given an integer vector (i, j, k) , this routine returns two integer vectors (i_1, j_1, k_1) and (i_2, j_2, k_2) satisfying $i_1 + j_1 + k_1 = E(n/2)$ (E =integer part), where $n = i + j + k$. This routine is used to know which powers we must multiply to get the power we want. This is used to obtain powers that are not in the common /pow/ (this happens with the powers of maximum order and during the filling of pow).

Subroutine MONOMI Given an integer vector (i, j, k) , a real number p , the Fourier series x, y and z and a working vector w containing a Fourier series, this routine returns $w = w + p \times x^i y^j z^k$, where w is the working vector. Before calling this routine a previous call to INIPOT (to compute the required powers of x, y and z) is needed.

Subroutine CORXET1 This routine computes the value between brackets ($[]$) in equation (1.1), by means of calls to MONOMI. The value is returned in expanded form inside a working vector.

Subroutine CORXET2 Like CORXET1, but for equation (1.2).

Subroutine CORXET3 Like CORXET1, but for equation (1.3).

Subroutine FUNC Routine that computes the value of function G (see page 4), calling routines CORXET1, CORXET2, CORXET3 and using the manipulator to compute the other terms appearing in the equations.

Subroutine DFUNC This routine computes the Jacobian of function G , calling the following routines:

- (1) D1X: it computes $\frac{\partial G_1}{\partial x}$.
- (2) D1Y: it computes $\frac{\partial G_1}{\partial y}$.
- (3) D1Z: it computes $\frac{\partial G_1}{\partial z}$.
- (4) D2X: it computes $\frac{\partial G_2}{\partial x}$.
- (5) D2Y: it computes $\frac{\partial G_2}{\partial y}$.
- (6) D2Z: it computes $\frac{\partial G_2}{\partial z}$.
- (7) D3X: it computes $\frac{\partial G_3}{\partial x}$.
- (8) D3Y: it computes $\frac{\partial G_3}{\partial y}$.
- (9) D3Z: it computes $\frac{\partial G_3}{\partial z}$.

Subroutine SISTEMA This routine solves a linear system by means of LU decomposition (see [9]). The operations are done taking into account that FORTRAN stores the matrices as a sequence of columns, and trying to avoid big jumps inside the matrix.

1.5 The Program

The program uses many data files in order to read the values of the basic frequencies (the values $\omega = (\omega_1, \dots, \omega_r)^t$), some phases related to them, control parameters (threshold to drop terms and to stop Newton method, maximum number of iterations and continuation parameters) and coefficients of the polynomial expansions of the equations. The program also reads a file containing the initial condition to start all the process (usually, it is taken equal to zero). During the execution, the program creates a file (called NEWTON.TMP) in which the actual solution is stored. This is due to the fact that if the new step of continuation does not succeed, the program can recover the last successful solution (for the last continuation parameter, of course) to start again with a smaller parameter. This file is usually small (about 20K). During the run, the program writes an execution log in a file called QPL4.CON. If the execution is successful, the result is stored in a file called QPL4.RES.

Subroutine INIEXP This routine reads the coefficients of the power expansion of the vector field from the files EXPAN.1, EXPAN.2 and EXPAN.3 for the first, second and third equations, respectively.

Subroutine LLEQ This routine reads one perturbing function P (from the data file QUS.COE), and multiplies the non-constant terms of it by the continuation parameter h .

Subroutine INQUS This routine reads all the perturbing functions stored in the file QUS.COE and multiplies the non-constant terms of them by the continuation parameter. To do this, the routine LLEQ is used.

Subroutine INIQP This routine initializes all the program. It reads the values of the basic frequencies (and phases) of the perturbations (from the file FREQUE.VAL), the control parameters for the Newton method (from the file CONTROL.QP), the coefficients of the power expansion of the vector field, the values of the perturbations and the initial condition (from QPL4.DAD). This routine calls INIEXP and INQUS.

Subroutine QAF This routine returns the number of terms of the residual acceleration with amplitude larger (in absolute value) than a given threshold.

Subroutine AFE This routine adds the terms of the residual acceleration whose amplitude is larger (in absolute value) than a given threshold to the set of variables and to the equations to be solved by Newton's method.

Subroutine CHECK This routine is called before adding terms to the Newton process in order to avoid having out-of-range arrays.

Subroutine CTI From the residual acceleration (the value of G), this routine computes the independent term of the linear system that appears in the Newton method. This step needs a "conversion" from Fourier series to an array, using the vectors of coefficients that take part in the Newton process.

Subroutine NOUP This routine computes the new approximation to the solution of $G = 0$, using the solution of the linear system. As in CTI, here we need

a “conversion” from an array to Fourier series.

Subroutine SC00BY This routine looks inside the components of the residual acceleration and searches for amplitudes larger (in absolute value) than a threshold `real*8 tol`, and adds them to the Newton process. The number of terms added must be between the `integer*4` values `mitn` and `matn`, modifying the value of `tol` if necessary. This routine calls the routine `AFE`.

Subroutine CNFNA This routine controls all the Newton process. It checks the actual step of the process and takes the following decisions:

- If the residual acceleration is not small enough, the new independent term is computed.
- If the value of G is small enough, it looks for “big terms” inside the residual acceleration (calling `SC00BY`), adding them if necessary.
- If G is small enough and there are not terms inside the residual acceleration big enough to be added to the process, the continuation parameter is moved.
- If the number of iterations with this value of the continuation parameter is too large, it sets a flag telling the routine `NWTA` to recover the last successful solution, and the continuation parameter step is halved.

The routine returns a flag containing what it has done, because the routine `NWTA` (this is the one that calls `CNFNA`) needs to know what is happening (see below).

Subroutine NWTA This routine performs the Newton method. It computes G and then calls `CNFNA` in order to know what to do. Depending on the value of the flag returned by this routine, the action taken can be:

- To continue with the process computing the Jacobian matrix (`DFUNC`), the independent term (`CTI`), solving the linear system (`SISTEMA`), obtaining the new approximation (`NOUP`) and computing again G .
- To overwrite the last successful solution stored in the disk file `NEWTON.TMP` with the solution obtained for the actual value of the parameter and to compute G again (the continuation parameter has been increased inside `CNFNA`).
- To recover the last successful solution from `NEWTON.TMP` and to compute G again (the continuation parameter has been moved inside `CNFNA`).
- To return the control to the main program because the final solution has been found.

Main Program Here we can find all the declarations of the arrays. This program calls `INIQP` to initialize the execution and then it calls `NWTA` to perform the Newton method. Finally, it opens the file `QPL4.RES` and writes in it the final result.

1.5.1 Test of the Program

In order to check all of these routines, we have solved a simplified version of the problem: it is considered planar and only the two most relevant frequencies are retained. They correspond to the mean elongation of the Sun (ψ) and to the difference between the mean longitude of the Moon and the mean longitude of the lunar perigee (M). The equations of motion are (see [1]):

$$\begin{aligned}\ddot{x} &= q_0 \left[-\frac{x - x_E}{r_{PE}^3} (1 - \mu) - \frac{x + x_E}{r_{PM}^3} \mu - x_E (1 - 2\mu) \right] + q_1 + q_2 x \\ &\quad + q_3 y + q_4 \dot{x} + q_5 \dot{y}, \\ \ddot{y} &= q_0 \left[-\frac{y - y_E}{r_{PE}^3} (1 - \mu) - \frac{y - y_E}{r_{PM}^3} \mu - y_E \right] + q_6 + q_7 y + q_8 \dot{x} + q_9 \dot{y},\end{aligned}$$

where x_E , y_E , r_{PE} and r_{PM} are defined as above, q_0, \dots, q_9 are functions of time:

$$q_i = \sum_{j=0}^6 A_{ij} \cos \theta_j + B_{ij} \sin \theta_j,$$

with $\theta_0 = 0$, $\theta_1 = 2\psi - M$, $\theta_2 = \psi$, $\theta_3 = M$, $\theta_4 = 2\psi$, $\theta_5 = 2M$ and $\theta_6 = 2\psi + M$. The values of ψ and M are defined by $\psi = 0.9251959855t + 5.0920835091$ and $M = 0.9915452215t + 2.2415337977$. We recall that the origin of time is the year 2000.0, and that 2π units of it are equivalent to a sidereal period of the Moon. The coefficients A_{ij} , B_{ij} are given in [1].

We start by setting the continuation parameter of the perturbation (h) to 1, and we select the degree of the power expansion equal to one. With this, we get a linear problem that the Newton method can solve easily. Then we add the quadratic terms (without need of any kind of continuation), and finally the third and fourth order terms. We have made another tests starting with the power expansion up to order four and moving the continuation parameter h from 0 to 1. All the results obtained with these tests are in good agreement with the ones of [1].

1.6 Results of the Algebraic Manipulator

We have done several runs of this software package. Initially, we tried to use the same scheme as in [1], that is, to fix the continuation parameter h to 1, and the degree n of the power expansion to 1. With this we got a linear problem that the Newton process solved without troubles. Then we added the second degree terms and by using continuation we could solve this problem, but the number of terms taking part in the Newton process was very large and we had many troubles with the memory and swap needed by the program. Then the third degree terms could not be added directly (this is because they are meaningful in this problem) and we had to use continuation again. During this continuation the number of terms

inside the Newton process became so high that we started to have troubles with the capacity of our computer. Nevertheless, the solution obtained by this method is good enough to be refined numerically by means of a parallel shooting (see [9]).

Due to the fact that this method does not seem to be the best to solve this problem (computer requirements are very high) we have implemented continuation for the perturbation (as it has already been mentioned during the description of the software). With this, we have obtained better results than the ones mentioned in the last paragraph. As an example, starting with the vector field expanded up to order six and giving to the Newton method a threshold $tol = 5 \times 10^{-5}$ (less than the accuracy of the model) to select the coefficients that will take part in the process, the program goes from continuation parameter $h = 0$ to $h = 1$ without any troubles, and the amount of time and memory needed are much less than before (this run is the one supplied as an example of the software).

Taking as initial condition the one supplied by this analytic approximation, we have integrated the model equations and Newton's equations of motion (using the JPL Ephemeris). In Figures 1.1 and 1.2 the solution given by the algebraic manipulator is compared with the integration of the model equations. The integration starts at the epoch 2000.0 and lasts for 1 year. Plots of the solution obtained integrating Newton's equations with the same initial conditions, for a time span of 2 and 5 years, are given in Figures 1.3 and 1.4 respectively. From now on, we shall call this the planar solution because it is close to the (x, y) -plane.

Note that after two years the solution is still contained in a domain of moderate size around L_4 , but after five years it has grown up to a rather large domain. This is due to the fact that the initial conditions are not good enough at this step and to the inherent instability of the motion around L_4 . Furthermore one can guess (specially in Figure 1.4, (x, z) projections) a sequence of Lissajous patterns. They suggest the existence of invariant unstable tori.

Now, if we try to improve the planar solution, supplying this solution as an initial condition for the manipulator, setting the continuation parameter h to 1, selecting the degree of the approximation to six (the program can handle up to degree nine, but the terms of degree larger than six can be neglected due to the actual size of the solution) and fixing the criterion to take terms inside the Newton process to $tol = 1 \times 10^{-5}$ (we stress that this improvement is relative to the model equations, not to the JPL model), we find out that, though the residual acceleration of the initial condition is very small, the method diverges. If we try to proceed as for the planar solution (starting from $h = 0$) but with $tol = 1 \times 10^{-5}$, we find out that the Newton process starts to decrease the continuation step until the program stops because the continuation step is too small (this happens for a continuation value near 0.98). Now if we take the intermediate solution in which the process has stopped as an initial condition, we set again $h = 0$ but we fix the value $tol = 1000$ (we do not allow more terms inside the Newton process than the ones of the initial condition) and we start again the program (with this we are dealing with a system

of algebraic equations of degree six and with fixed dimension) we find that near the point $h = 0.98$ the Jacobian of G has a change of sign. This is related to some bifurcation or turning point.

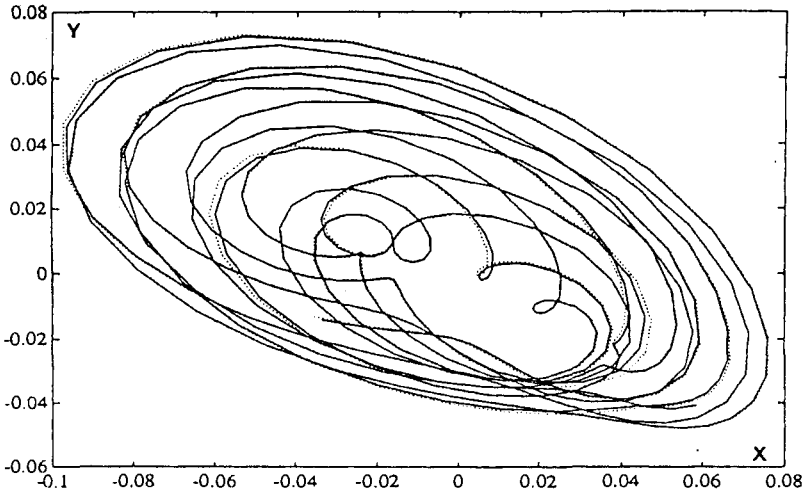


Fig. 1.1 (x, y) projection of the solution obtained integrating the model equations (dotted line) and the values of the solution provided by the manipulator (solid line).

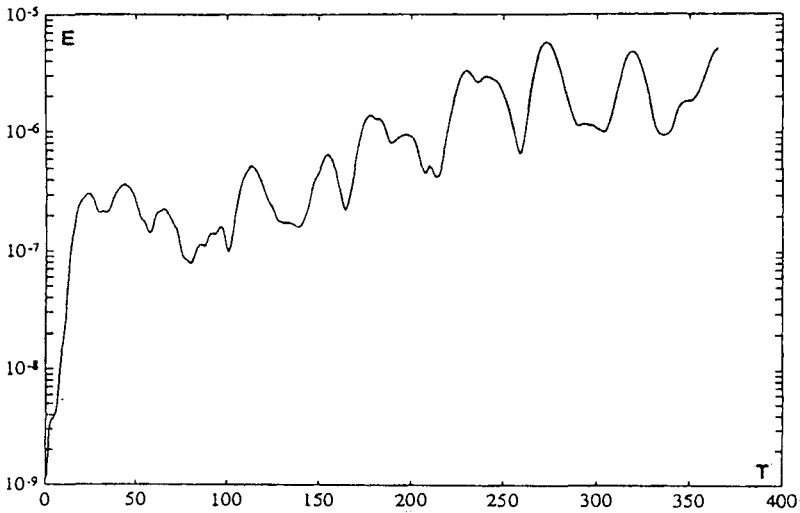


Fig. 1.2 Euclidean norm of the differences (in the phase space) between the solution obtained by integration and the one given by the manipulator.

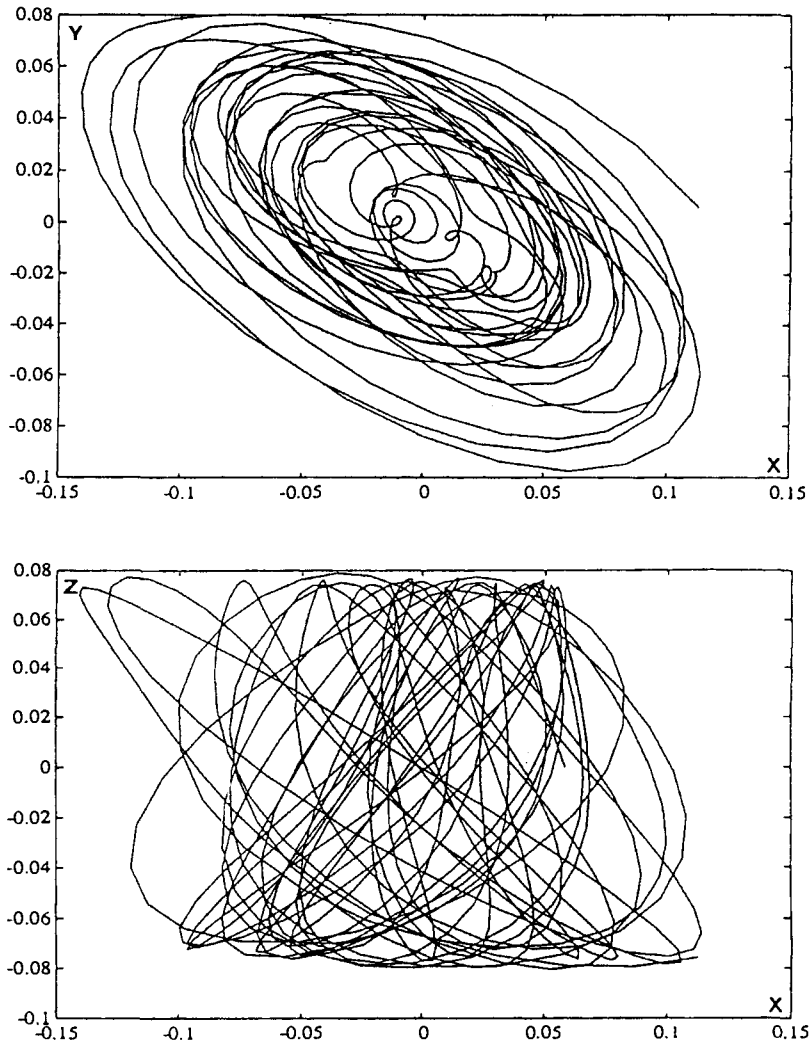


Fig. 1.3 Projections of the solution obtained integrating Newton's equations of motion for a time span of 2 years. The initial condition is the one given by the manipulator and it has not been refined.

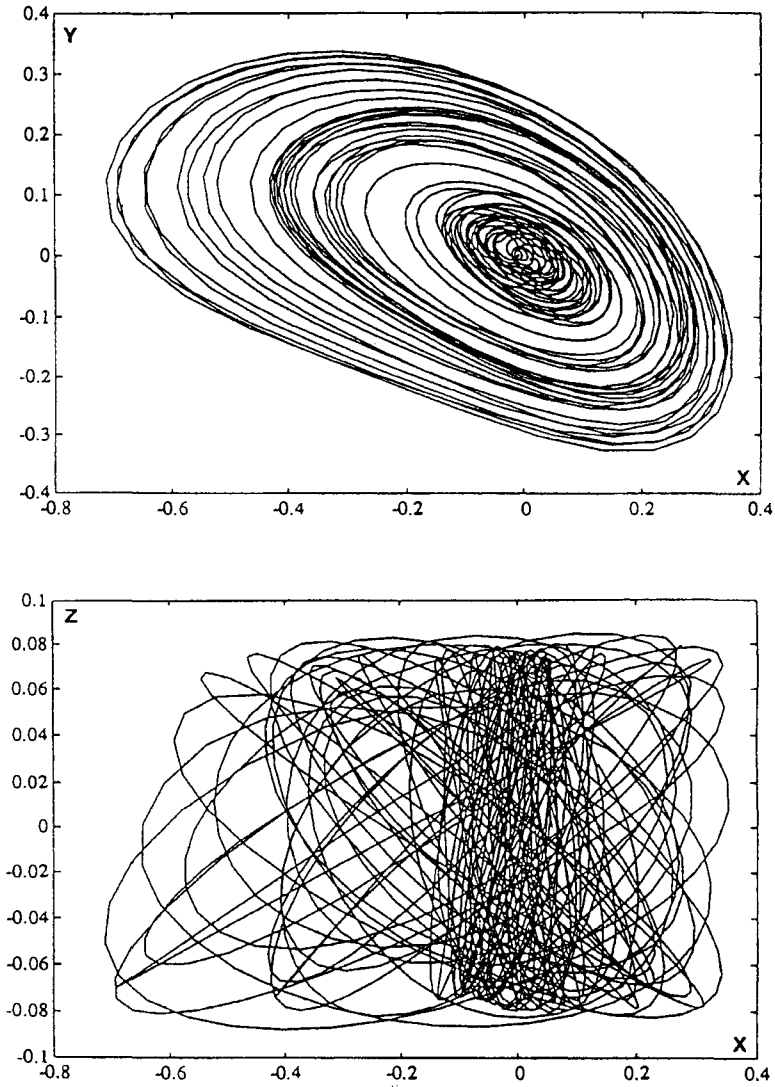


Fig. 1.4 Same as Figure 1.3 but for a time span of 5 years.

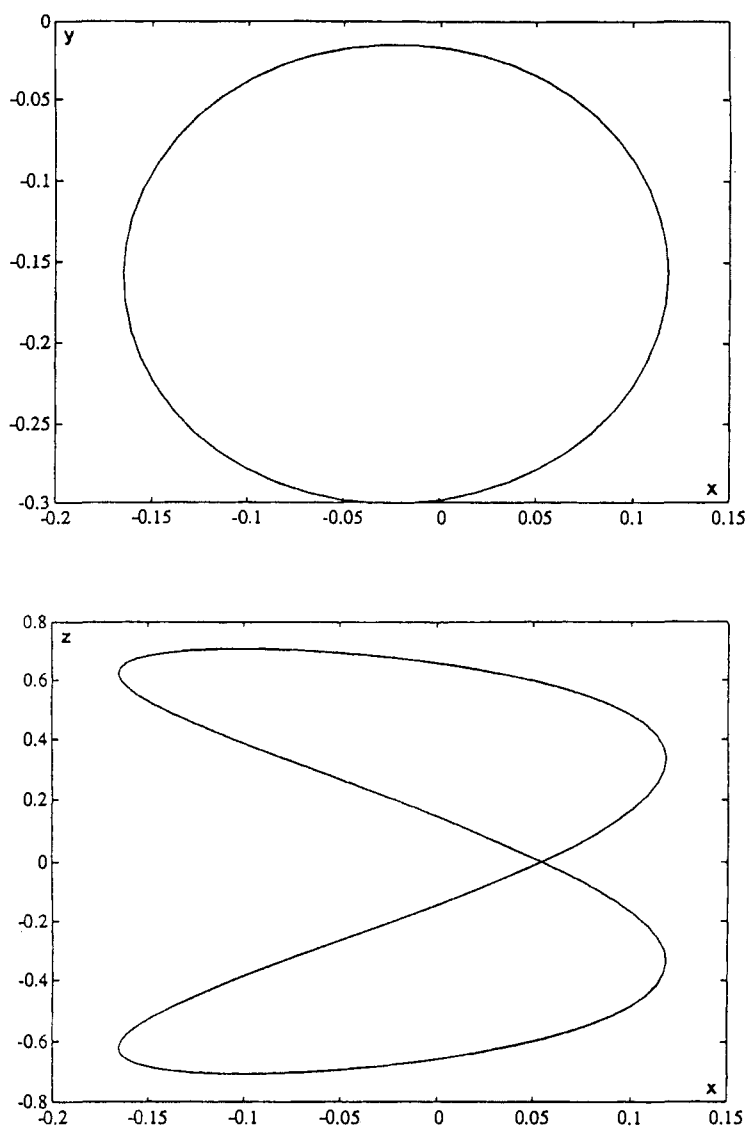


Fig. 1.5 Projection of the periodic orbit of averaged model equations.

This last point seems related to the following fact: take the model equations and average them with respect to time. Then, this autonomous system has a periodic solution (see Figure 1.5) whose frequency is rather close to the difference between the first and the third basic frequencies of the model equations (they are, respectively, the mean longitude of the Moon (equal to one, because of the choice of the units) and the mean longitude of the ascending node of the Moon). This orbit seems to be related to the vertical oscillations of the RTBP. When the non-constant part of the perturbation is added, this periodic orbit bifurcates to a quasi-periodic orbit having the same basic frequencies than the ones of the perturbation. This leads us to the fact that the problem of finding a quasi-periodic solution near L_4 has lots of solutions, but probably only a few of them have as basic set of frequencies the one of the excitation. Furthermore only a part of the last ones can be obtained as a natural continuation of L_4 .

1.7 Numerical Refinement

In order to obtain a good nominal orbit we have implemented a parallel shooting method (see [9]) to get a solution of the JPL model very similar to the one found with the algebraic manipulator.

Now, let us see some details about this parallel shooting. First of all, we split the time span $[a, b]$ in which we want to find the nominal orbit in several pieces $[t_i, t_{i+1}]$, $i \in \{0, \dots, n-1\}$, verifying that $t_0 = a$, $t_n = b$ and $h = t_{i+1} - t_i = (b-a)/n$ (different time intervals can be used but we have chosen here all of them equal). Now let i be a value between 1 and $n-1$ and let \mathcal{F}_i be the function defined as follows: if x_i is a point of the phase space, then $\mathcal{F}_i(x_i) = y_{i+1}$, where y_{i+1} is the value of the solution defined by the initial conditions (t_i, x_i) at time t_{i+1} , under the flow of the JPL model. Now we consider the vector (x_0, \dots, x_n) , where x_i are coordinates in the phase space corresponding to time t_i . We can define $\mathcal{F} = (\mathcal{F}_0, \dots, \mathcal{F}_{n-1})$ in the usual way:

$$(x_0, \dots, x_n) \xrightarrow{\mathcal{F}} (y_1, \dots, y_n),$$

and note that, if the values x_i are all from the same solution of the JPL model, then we should have that $y_i = x_i$, $1 \leq i \leq n$. Then we impose these conditions and this leads us to solve a nonlinear system of $6n$ equations and $6n + 6$ unknowns. To do this, we have used again a Newton method and we have taken as initial guess the values provided by the planar solution obtained using the algebraic manipulator. Due to the fact that we have more unknowns than equations, we have added six more conditions: we have fixed the initial position (the three first components of x_0) and the final one (the three first components of x_n). With this, the system we deal with has $6n$ equations and the same number of unknowns. Finally, we note that the linear system that appears at each iteration of the Newton method

has a band matrix. For this reason we have written a routine adapted to this kind of matrix. Another way to perform the parallel shooting procedure can be the following: instead of adding to the set of $6n$ equations the above mentioned six conditions (which in some sense can force the solution in an nonnatural way), replace them trying to minimize, with respect to some norm, the total corrections to be done at each step. This procedure has not been implemented.

Subroutine SISBAN This routine solves a linear system of equations with a banded matrix, using only the band of the matrix plus three diagonals. The method is based in Gaussian elimination with partial pivoting.

Subroutine OMPP This routine computes the initial guess for the Newton method, from a Fourier series corresponding to an approximation obtained by means of the algebraic manipulator. This guess is stored in a vector. This routine uses the routines AVALUA and DF of the manipulator.

Subroutine COIN This routine changes coordinates, from the normalized ones to ecliptic ones (JPL), to the vector of initial conditions obtained by OMPP.

Subroutine FUN This routine computes the value of the function we want to make zero. For this purpose it performs a numerical integration of the equations of motion in the solar system using the JPL tapes to obtain the position of the bodies. The routine also integrates the variational equations and fills the Jacobian matrix of the function. The device used in this numerical integration has been the Shampine-Gordon method (see [8]).

Subroutine COOF This routine performs a change of coordinates, from the ecliptic ones to normalized ones, to the vector of final conditions obtained by the Newton method.

Main Program This routine reads the basic frequencies of the perturbations from the file FREQUE.VAL, the initial Fourier series (to obtain the initial condition for the Newton method) from the file QPL4.RES, some control parameters from the file TPL4.CTL (initial time, time step, number of time steps for the Newton process as well as thresholds for the numerical integration and a tolerance to know when the Newton process has finished) and the model of the solar system used from the file MODEL.DAT. With this, the program performs a Newton method (using the routines mentioned above), writing in the console some information about what it is doing. When the solution is found, the program writes the initial condition of the final orbit in the file TPL4.RES.

1.7.1 *Final Results*

Now, using the program described above, it is possible to find an orbit for the JPL model with a behavior similar to the one of the planar solution obtained by the manipulator. In Figure 1.6 we show the numerical refinement of the planar orbit obtained in Section 1.6 for a time span of five years starting at the year 2000. The

time step used in the parallel shooting has been of 1 day. To refine this orbit for longer time spans is very difficult (the parallel shooting does not converge) due to the fact that there exists a resonance (already mentioned in Section 1.6) implying that this solution is not an exact solution of the system but a minimum of the residual acceleration. Nevertheless, the solution found for five years has a very slow “diffusion” and it goes away of L_4 slowly. As an example of this, Figure 1.7 shows its behavior for more than seven years, and it looks like quasi-periodic.

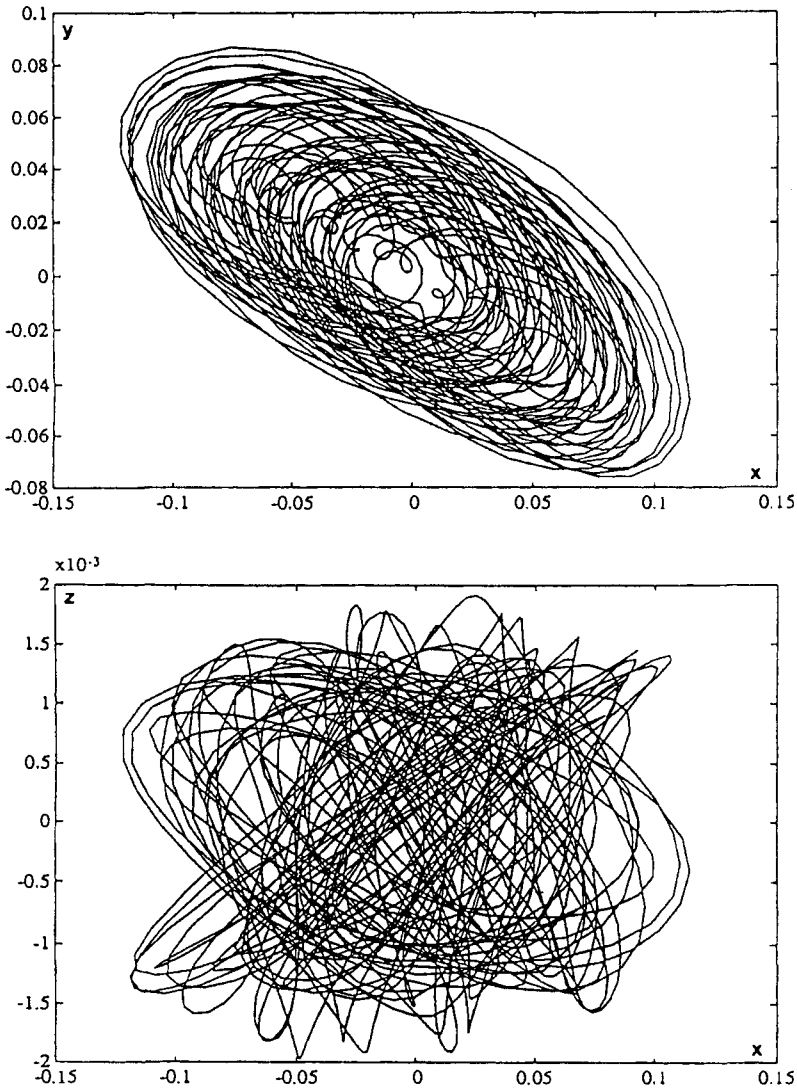


Fig. 1.6 Numerical refinement of the planar orbit for a time span of 5 years.

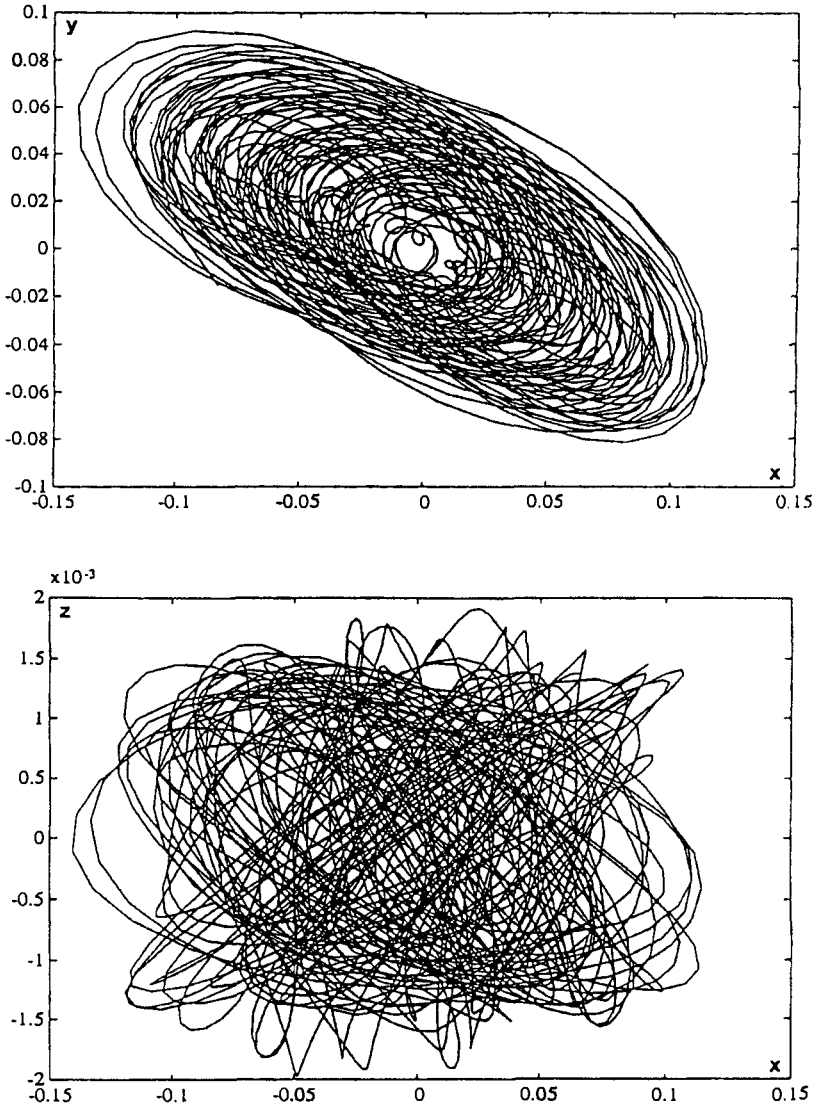


Fig. 1.7 The same orbit of Figure 1.6 but integrated for a larger time span, which goes from 500 days before 2000.0 to 2100 days after 2000.0.

1.8 The Neighbourhood of the Computed Nearly Quasi-periodic Solution

To study both the stability properties of the computed solution and the relative motion of particles running in orbits close to that one, we have performed a variational study. Of course, in this case we can not talk about monodromy matrix because we never return to the initial conditions or, even if this happens, the subsequent motion will be different. But we can look for points returning close to the initial conditions on the phase space. Figure 1.8 gives the distance to the initial conditions in adimensional units as a function of time. One can see a typical quasi-periodic pattern and after, roughly, 1625 days, we are not too far from the initial point. Now we can consider the variational matrix after that time interval as if it were a monodromy matrix.

On the other hand, during the integration of the variational equations (using the JPL model, of course), we can record the spectral radius of the instantaneous variational matrix. That value has been plotted as a function of time in Figure 1.9, during the full period 2000.0 to 2005.0. Letting aside some local (in time) irregularities on the first 600 days, the pattern is quite neat. These irregularities can be due to the strong constraints on the parallel shooting method which forces the initial and final positions to coincide with the one given by the semianalytic approximation. From the end of the second year on, there are large oscillations with half a year period with superimposed smaller oscillations with period roughly equal to one half of the lunar synodic period. This can be better seen in Figure 1.10, where we have plotted the quotient $\ln \rho(t)/t$. Here t is the time elapsed since the initial epoch 2000.0 and $\rho(t)$ is the spectral radius (i.e. the maximum of the moduli of the eigenvalues) of the variational matrix at the current time. To skip the initial irregular behavior this has been plotted starting at the year 2002.0 and the abscissae in the figure denote days since 2002.0. The limit of $\ln \rho(t)/t$ should give the maximal Lyapunov exponent. This seems to be close to 3.65×10^{-3} if the time unit is one day. This implies an average increasing of the distance to the computed orbit at a rate of $\exp(365 \times 3.65 \times 10^{-3}) \approx 3.79$ per year. This is an extremely mild instability and offers no problems concerning station keeping.

We return to the approximate monodromy matrix after 1625 days. Let λ_i , $i = 1, \dots, 6$ the eigenvalues of this matrix. The numerical values are

$$\begin{aligned} \lambda_1 &= -30.664197, & \lambda_3 &= -0.079456 + 0.996838i, & \lambda_5 &= -0.511657, \\ \lambda_2 &= -1.954433, & \lambda_4 &= -0.079456 - 0.996838i, & \lambda_6 &= -0.032611. \end{aligned}$$

They appear almost exactly in reciprocal pairs ($\lambda_j \lambda_{7-j} = 1$) as it happens for autonomous Hamiltonians. Note that the chosen epoch (2000.0 + 1625 days) is particularly good. For that epoch $\ln \rho(t)/t$ equals 2.11×10^{-3} . But we recall that the limit value 3.65×10^{-3} , say, is an average. The eigenvalue λ_1 (resp. λ_6) is related to a mild instability (resp. stability). Then λ_2 (resp. λ_5) to a very mild instability

e_1	e_2	e_3, e_4	e_5	e_6
-0.163767	-0.286769	$0.169256 \pm -0.395523i$	0.321951	0.375073
0.964713	0.940385	$0.946713 \pm 0.000000i$	0.930086	0.904416
0.072496	0.053245	$-0.110340 \pm 0.913048i$	-0.033994	-0.074290
0.189793	0.162855	$0.208013 \pm 0.002104i$	0.160132	0.176994
-0.032347	-0.059263	$0.028111 \pm -0.093038i$	0.061770	0.064810
-0.013662	-0.023880	$0.137305 \pm 0.035283i$	-0.025978	-0.017493

Table 1.1 Eigenvectors of the approximate monodromy matrix.

(resp. stability), and λ_3, λ_4 are related to neutral behavior (on the average during that time interval). The corresponding eigenvalues, e_1 to e_6 , are given in Table 1.1 with modulus normalized to 1.

We note that the neutral modes have a stronger relation with the vertical direction (z, \dot{z}) than the other ones. This is related to the suspected resonance between the proper vertical frequency of the restricted problem around L_4 and some of the exciting frequencies.

To see the behavior of a nearby particle we have computed the following vectors: let $A(t)$ the solution of the variational equations starting with $A(0) = \text{Id}$, where time is counted since the epoch 2000.0. Then $e_j(t) = A(t)e_j$, $j = 1, \dots, 6$ will give the relative evolution of a particle with respect to the basic orbit (or of one particle with respect to another one if they are located initially at $p \pm \alpha e_j$, where p is the initial condition corresponding to the quasi-periodic solution and α is a small quantity). The vectors $e_j(t)$, $j = 1, \dots, 6$, with $e_j(0)$ of modulus 1, have been projected on the (x, y) -plane. Figures 1.11, 1.12, 1.13, 1.14, 1.15 and 1.16 present the results for the modes $e_1(t)$ to $e_6(t)$ respectively. To see the dominant exponentially increasing (resp. decreasing) character of the modulus of $e_1(t)$ (resp. $e_6(t)$) we present the Figures 1.11 and 1.16. Figure 1.11 has been obtained as follows: let $(r(t), \theta(t))$ be the polar coordinates of the (x, y) projection of $e_1(t)$. Compute $\rho(t) = \text{argsinh}(r(t))$ and plot the polar coordinates $(\rho(t), \theta(t))$. Figure 1.16 has been obtained with the same procedure that Figure 1.11, but with the scaling $\rho(t) = \text{argsinh}(20r(t))/\text{argsinh}(20)$. Figures 1.12, 1.13, 1.14 and 1.15 have not any kind of scaling. In Figures 1.11 and 1.16 one can see the superposition of two dominant periodic contributions added to the exponential behavior.

Finally if we consider two nearby particles located at symmetrical positions with respect to the quasi-periodic solutions, we suggest the possibility of putting them along the unstable direction. This is because the line joining the particles is sweeping all the directions close to the ecliptic. Assume the relative distance is initially 1 km. Then, after one month, their distance will increase, on the average, by $\exp(30 \ln(\lambda_1)/1625) - 1 \approx 0.065$ km.

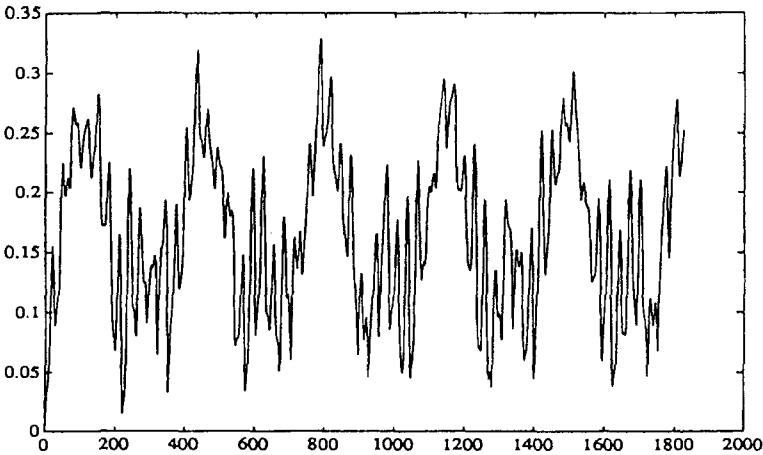


Fig. 1.8 Distance to the initial conditions in adimensional units as a function of time (in days) from 2000.0 to 2005.0.

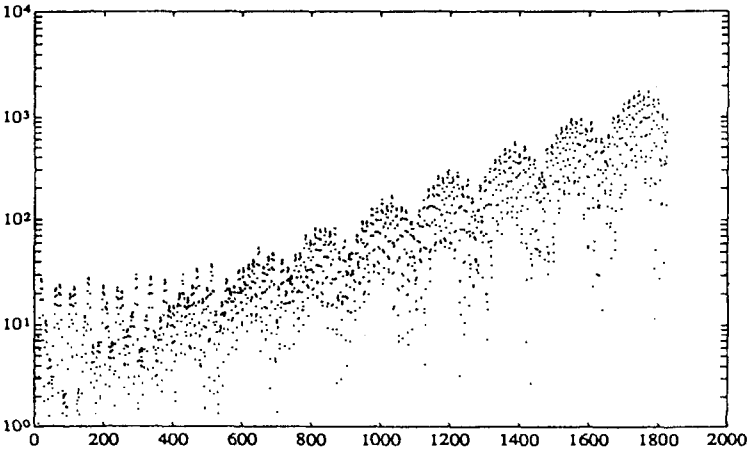


Fig. 1.9 Spectral radius of the variational matrix as a function of time (in days) from 2000.0 to 2005.0.

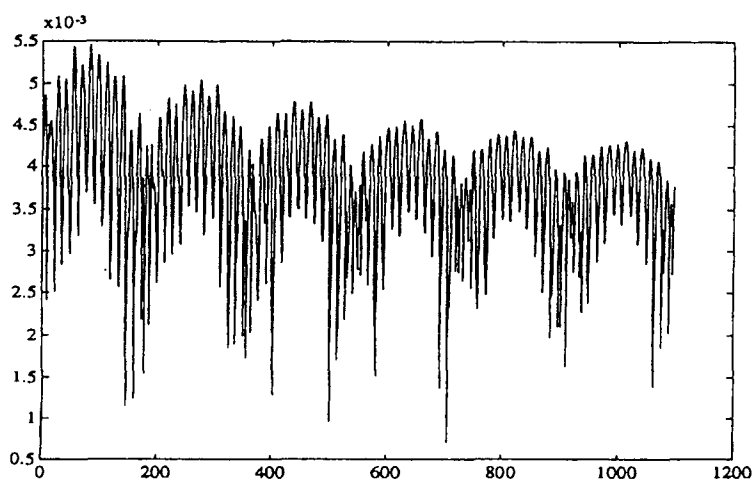


Fig. 1.10 Values of $\ln \rho(t)/t$, where the time t (in days) ranges from 2002.0 to 2005, and $\rho(t)$ is the spectral radius of the variational matrix.

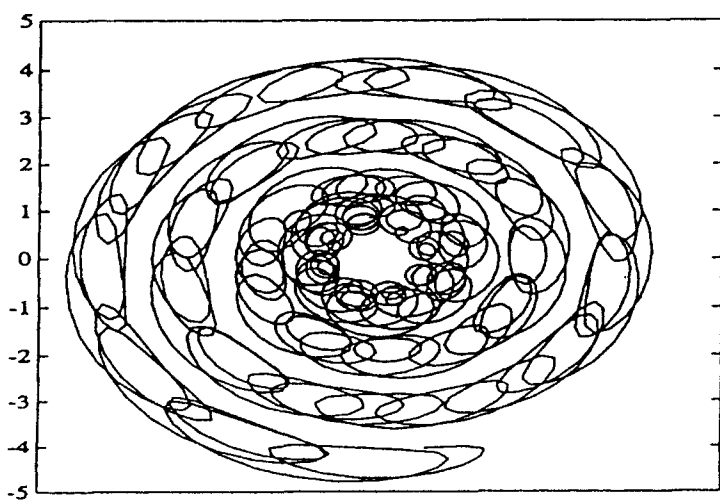


Fig. 1.11 (x, y) projection of the mode $e_1(t)$. Polar coordinates have been used, with the scaling $r \sim \operatorname{argsinh}(r)$. Time span: 1625 days.

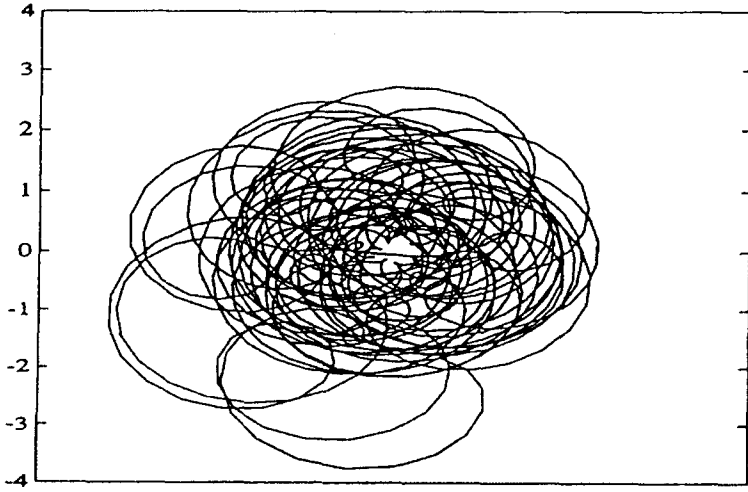


Fig. 1.12 (x, y) projection of the mode $e_2(t)$. Time span: 1625 days.

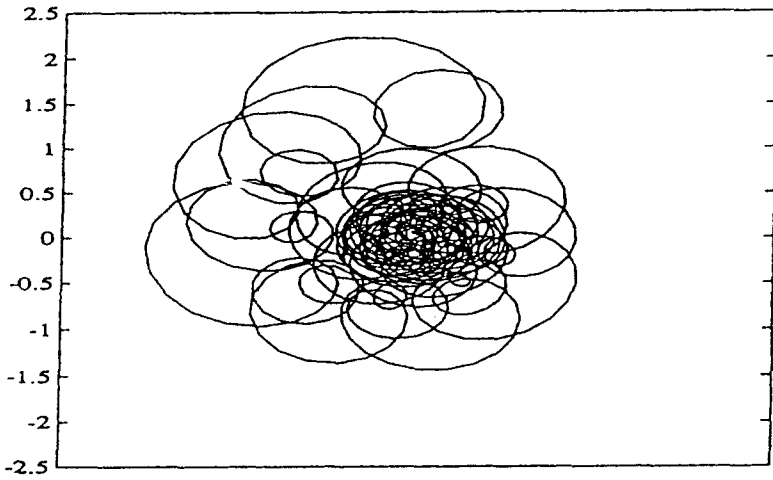


Fig. 1.13 (x, y) projection of the real part of the mode $e_3(t)$. (Note that $\text{Re}(e_3) = \text{Re}(e_4)$). Time span: 1625 days.

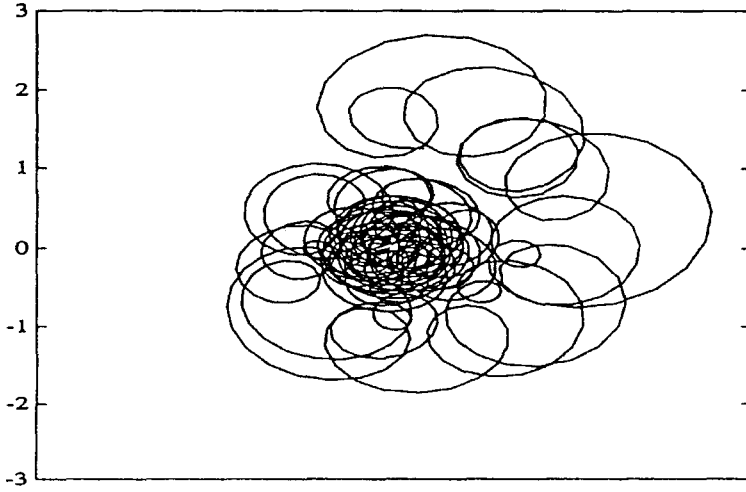


Fig. 1.14 (x, y) projection of the imaginary part of the mode $e_3(t)$. (Note that $\text{Im}(e_3) = -\text{Im}(e_4)$). Time span: 1625 days.

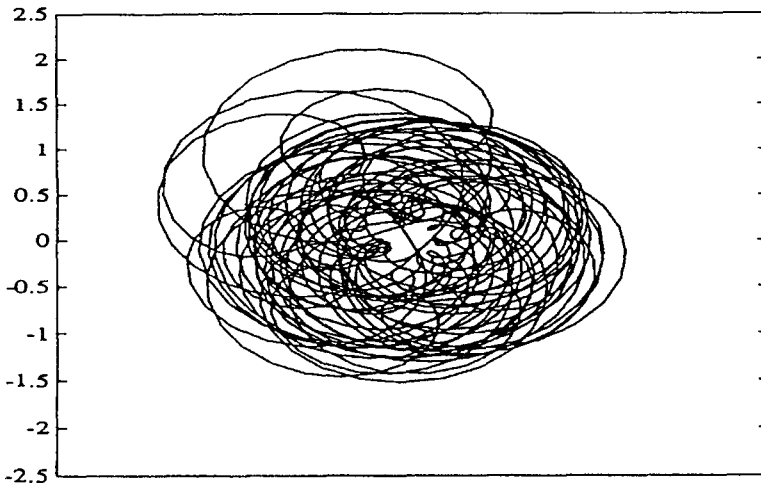


Fig. 1.15 (x, y) projection of the mode $e_5(t)$. Time span: 1625 days.

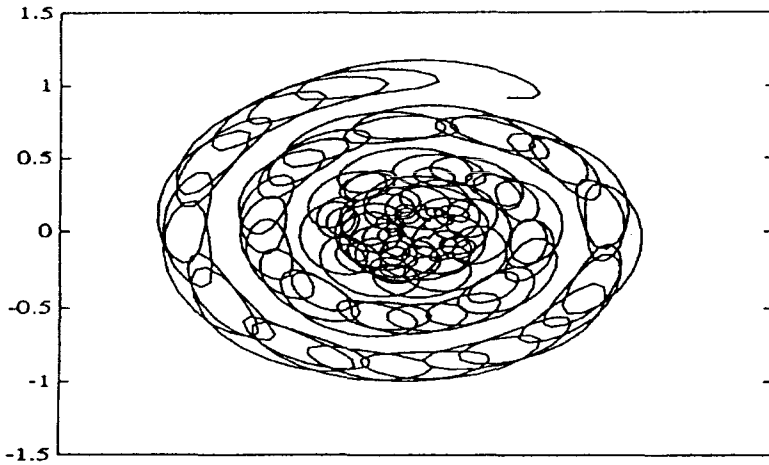


Fig. 1.16 (x, y) projection of the mode $e_6(t)$. Polar coordinates have been used, with the scaling $r \sim \operatorname{argsinh}(20r)/\operatorname{argsinh}(20)$. Time span: 1625 days.

1.9 Problems and Extensions

Looking for the behavior of the orbits near the geometrical triangular points of the Earth–Moon system we are faced with several problems. If we consider first the RTBP the phase space is filled up by periodic orbits, two-dimensional and three-dimensional tori as well as stochastic zones. The last ones are extremely small near the equilibrium point but the size increases when the distance to the equilibrium point does. To this simplified model we add relatively very important perturbations, mainly due to the effect of the Sun and to the noncircular motion of the Moon and the Earth. Those perturbations can be assumed to be quasi-periodic for moderate time intervals. This produces strong resonances in some domains of the phase space and therefore large deviations from the (relatively) clear situation found for the RTBP. For small perturbations we have just to add a kind of quasi-periodic swinging to the previous structure. But the actual size of the perturbations is big enough to produce bifurcations loosing uniqueness. Therefore this is a source of problems for the analytic approximate solutions and for the subsequent numerical refinement as stated in 1.6 and 1.7.

The global problem, that is, the study with great detail of the real behavior in a full neighborhood of the former equilibrium point is worthy. It is useful for

the design of future missions. From the abstract point of view there is a lack of supporting theory. From the analytic and numerical points of view this is a formidable task. We think that the methodology to be used to deal with this problem would be useful in many other contexts.

Chapter 2

Global Description of the Orbits Near the L_1 Point of the Earth–Sun System in the RTBP

2.1 Introduction

In the spatial RTBP the L_1 and L_2 libration points have a linear behavior of the type *saddle* \times *center* \times *center*. Under generic assumptions there exist nearby two-dimensional tori. Because of the small divisors problems, on those tori the frequencies should satisfy a Diophantine condition. But as we are interested in short time intervals the small divisors problems do not show up unless they are associated to low order resonances. Let ω_1, ω_2 be the two basic frequencies at L_i , $i=1, 2$. For small values of μ the frequencies are rather close (for the Hill's problem, $\mu = 0$, they are $(28^{1/2} - 1)^{1/2}$ and 2, respectively). When the amplitudes of the orbit change so do the frequencies. It is known that the Lyapunov orbits bifurcate to halo orbits when the x -amplitude (x is the axis with origin at L_i in the direction of the primaries) is roughly 0.139 if the unit of distance is the one from the small mass to the libration point. Hence this is the distance corresponding to the 1 to 1 resonance giving rise to the halo orbits. If we keep our study in a smaller region the effect of this resonance is not too strong. Other resonances are found to higher order. As $\omega_1 / \omega_2 \simeq 29/28$ for the Hill's problem, the next resonances to play a significant role are of order 57. They are of small amplitude and long period. Therefore we can try to obtain the invariant tori formally to some order (less than 57, of course) and taking only into account the 1 to 1 resonance. Cutting the expressions to this order we obtain a foliation by invariant tori in some region around the equilibrium points. For some of the amplitudes the tori certainly do not exist, but the small size of the stochastic zone makes this fact irrelevant for short time interval applications, in our case a spacecraft running during a few years.

2.2 The Equations of Motion

We restrict the exposition to the L_1 libration point (using the convention that this relative equilibrium point is located between the primaries). We consider the

following system of reference: The positive x -axis is directed from L_1 to the smaller primary. The y -axis is in the plane of sidereal motion of the primaries at $\pi/2$ from the x -axis in counterclockwise sense. The z -axis completes a positively oriented coordinate system. Let $\gamma = d(m_2, L_1)$, where m_2 denotes the position of the smaller primary. The origin is taken at the libration point and γ is the unit of distance. The unit of time is such that the period of revolution of the primaries in the sidereal system is 2π . Let μ be the mass of the smaller primary (in our case the Earth, or rather the Earth-Moon barycenter; $\mu \approx 0.3040357143 \times 10^{-5}$ in this chapter) and $1 - \mu$ the one of the bigger primary (the Sun in our application). The value of γ is obtained from the Euler quintic equation:

$$\gamma^5 - (3 - \mu)\gamma^4 + (3 - 2\mu)\gamma^3 - \mu\gamma^2 + 2\mu\gamma - \mu = 0.$$

It can be solved by Newton method starting at $(\mu/3)^{1/3}$.

Let X, Y, Z be the usual synodic coordinates centered at the center of mass. The equations of motion are

$$\begin{aligned} \ddot{X} - 2\dot{Y} &= \Omega_X, \\ \dot{Y} + 2\dot{X} &= \Omega_Y, \\ \ddot{Z} &= \Omega_Z, \end{aligned} \tag{2.1}$$

where

$$\Omega = \frac{1}{2}(X^2 + Y^2) + (1 - \mu)r_1^{-1} + \mu r_2^{-1},$$

being

$$\begin{aligned} r_1^2 &= (X - \mu)^2 + Y^2 + Z^2, \\ r_2^2 &= (X - \mu + 1)^2 + Y^2 + Z^2. \end{aligned}$$

As we have described we make the change of coordinates

$$\begin{aligned} x &= -\frac{1}{\gamma}(X - \mu + 1 - \gamma), \\ y &= -\frac{1}{\gamma}Y, \\ z &= \frac{1}{\gamma}Z. \end{aligned} \tag{2.2}$$

Then

$$\begin{aligned} r_1^{-1} &= (1 - \gamma)^{-1} \left(1 + \frac{2\gamma}{1 - \gamma}x + \frac{\gamma^2}{(1 - \gamma)^2}\rho^2 \right)^{-1/2} \\ &= (1 - \gamma)^{-1} \sum_{n \geq 0} \left(\frac{\gamma\rho}{1 - \gamma} \right)^n (-1)^n P_n \left(\frac{x}{\rho} \right), \end{aligned}$$

$$r_2^{-1} = \gamma^{-1}(1 - 2x + \rho^2)^{-1/2} = \gamma^{-1} \sum_{n \geq 0} \rho^n P_n \left(\frac{x}{\rho} \right),$$

where

$$\rho^2 = x^2 + y^2 + z^2$$

and P_n denote the Legendre polynomials.

Let

$$c_n = \gamma^{-3} \left(\mu + (-1)^n (1 - \mu) \left(\frac{\gamma}{1 - \gamma} \right)^{n+1} \right). \quad (2.3)$$

Then the equations of motion are written as

$$\begin{aligned} \ddot{x} - 2\dot{y} - (1 + 2c_2)x &= \frac{\partial}{\partial x} \sum_{n \geq 3} c_n \rho^n P_n \left(\frac{x}{\rho} \right), \\ \ddot{y} + 2\dot{x} + (c_2 - 1)y &= \frac{\partial}{\partial y} \sum_{n \geq 3} c_n \rho^n P_n \left(\frac{x}{\rho} \right), \\ \ddot{z} + c_2 z &= \frac{\partial}{\partial z} \sum_{n \geq 3} c_n \rho^n P_n \left(\frac{x}{\rho} \right). \end{aligned} \quad (2.4)$$

We recall also, for further use, that (2.1) can be written in Hamiltonian form. Introducing

$$\begin{aligned} P_X &= \dot{X} - Y, \\ P_Y &= \dot{Y} + X, \\ P_Z &= \dot{Z}, \end{aligned}$$

the Hamiltonian is

$$\begin{aligned} H &= \frac{1}{2}(P_X^2 + P_Y^2 + P_Z^2) + YP_X - XP_Y - (1 - \mu)r_1^{-1} - \mu r_2^{-1} \\ &= \frac{1}{2}(\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2) - \Omega. \end{aligned} \quad (2.5)$$

Let $T_n = \rho^n P_n(x/\rho)$, $R_{n-1} = y^{-1} \partial T_{n+1} / \partial y = z^{-1} \partial T_{n+1} / \partial z$. Then (2.4) gives

$$\begin{aligned} \ddot{x} - 2\dot{y} - (1 + 2c_2)x &= \sum_{n \geq 2} c_{n+1} (n+1) T_n, \\ \ddot{y} + 2\dot{x} + (c_2 - 1)y &= y \sum_{n \geq 2} c_{n+1} R_{n-1}, \\ \ddot{z} + c_2 z &= z \sum_{n \geq 2} c_{n+1} R_{n-1}. \end{aligned} \quad (2.6)$$

We remark that T_n (resp. R_{n-1}) is a polynomial of degree n (resp. $n-1$), in x , y and z .

The equations 2.6 can also be put in Hamiltonian form. Let $p_x = \dot{x} - y$, $p_y = \dot{y} + x$, $p_z = \dot{z}$. Then the corresponding Hamiltonian is

$$K = \frac{1}{2} (p_x^2 + p_y^2 + p_z^2) + yp_x - xp_y - \sum_{n \geq 2} c_n \rho^n P_n \left(\frac{x}{\rho} \right). \quad (2.7)$$

The Hamiltonians H and K are related by

$$H = K\gamma^2 - \frac{1}{2}(1 - \gamma - \mu)^2 - \frac{\mu}{\gamma} - \frac{1 - \mu}{1 - \gamma}. \quad (2.8)$$

2.3 Formal Series Solutions

If we skip in (2.6) the right-hand side terms (of degree ≥ 2) the solution of the system, restricted to the center manifold (i.e. deleting the exponentially increasing or decreasing terms) is

$$\begin{aligned} x &= \alpha \cos(\omega_1 \tau + \phi_1), \\ y &= -\bar{k} \alpha \sin(\omega_1 \tau + \phi_1), \\ z &= \beta \cos(\omega_2 \tau + \phi_2), \end{aligned} \quad (2.9)$$

with

$$\bar{k} = (\omega_1^2 + 2c_2 + 1)/(2\omega_1).$$

The parameters α, β are arbitrary amplitudes and ϕ_1, ϕ_2 are arbitrary phases. The frequencies in (2.9) are $\omega_1 = ((2 - c_2 + (9c_2^2 - 8c_2)^{1/2})/2)^{1/2}$, $\omega_2 = c_2^{1/2}$. As (2.6) is an autonomous system we can take the origin of time such that $\phi_1 = 0$. Furthermore we denote $\omega_1 \tau$ and $\omega_2 \tau + \phi_2$ as θ_1 and θ_2 , respectively. When the right-hand side terms of (4) are considered we should allow for varying frequencies depending on the amplitudes, according to the Lindstedt-Poincaré method to avoid secular terms. Then θ_1 will be now $d\tau$ and θ_2 will be $f\tau + \phi_2$, where $d = \omega_1 + O(\alpha, \beta)$, $f = \omega_2 + O(\alpha, \beta)$.

We look for solutions in complex exponential form of the type

$$\begin{aligned} x &= \sum x_{i,j,k,m} \alpha^i \beta^j \gamma_1^k \gamma_2^m, \\ y &= \sqrt{-1} \sum y_{i,j,k,m} \alpha^i \beta^j \gamma_1^k \gamma_2^m, \\ z &= \sum z_{i,j,k,m} \alpha^i \beta^j \gamma_1^k \gamma_2^m, \end{aligned} \quad (2.10)$$

where $\gamma_s = \exp(\sqrt{-1}\theta_s)$, $s = 1, 2$. Due to the symmetries of the problem the coefficients in (2.10) satisfy the relations:

- (1) $i, j \geq 0, i + j \geq 1, |k| \leq i, |m| \leq j, i - k \equiv 0 \pmod{2}, j - m \equiv 0 \pmod{2}$
in all the cases,

- (2) $x_{i,j,-k,-m} = x_{i,j,k,m}$, $y_{i,j,-k,-m} = -y_{i,j,k,m}$, $z_{i,j,-k,-m} = z_{i,j,k,m}$. Hence it is enough, for the computations, to keep only the terms with $k > 0$ or, if $k = 0$, with $m \geq 0$,
- (3) For x and y one should have $j \equiv 0 \pmod{2}$, and for z one should have $j \equiv 1 \pmod{2}$.
- (4) The Lindstedt-Poincaré method can be normalized in such a way that for $(k, m) = (1, 0)$ the only term appearing in x corresponds to $(i, j) = (1, 0)$. In an analogous way if $(k, m) = (0, 1)$ the only term in z corresponds to $(i, j) = (0, 1)$.
- (5) In d and f there are only terms of the form $\alpha^i \beta^j$ with i and j even.

A sum as x (resp. y , resp. z) will be called of type \underline{a} (resp. \underline{b} , resp. \underline{c}). The product of a series of type \underline{a} by another of type \underline{a} , \underline{b} or \underline{c} gives again a series of type \underline{a} , \underline{b} or \underline{c} , respectively. The product of a series of type \underline{b} by another of type \underline{b} (or of one of type \underline{c} by another of type \underline{c}) gives again a series of type \underline{a} . The series \dot{x} , \ddot{x} , \dot{y} , \ddot{y} and \ddot{z} , appearing in (2.6) are, respectively of types \underline{b} , \underline{a} , \underline{a} , \underline{b} and \underline{c} .

The computation of the terms in (2.10) is done by increasing order of $n = i + j$. For $n = 1$ one has from (2.9) the values

$$x_{1,0,1,0} = 1/2, \quad y_{1,0,1,0} = \bar{k}/2, \quad z_{0,1,0,1} = 1/2.$$

The terms T_m and R_m in (2.6) are obtained by a recurrence which follows from the one of the Legendre polynomials

$$\begin{aligned} T_m(x, y, z) &= \left(2 - \frac{1}{m}\right) x T_{m-1} - \left(1 - \frac{1}{m}\right) \rho^2 T_{m-2}, \\ R_m(x, y, z) &= \frac{2m+3}{m+2} x R_{m-1} - \frac{2m+2}{m+2} T_m - \frac{m+1}{m+2} \rho^2 R_{m-2}, \end{aligned} \quad (2.11)$$

starting with $T_0 = 1$, $T_1 = x$, $R_0 = -1$, $R_1 = -3x$.

The derivatives with respect to τ are obtained by means of

$$\frac{d}{d\tau} = d \frac{\partial}{\partial \theta_1} + f \frac{\partial}{\partial \theta_2}.$$

Then \dot{x} , \ddot{x} , \dot{y} , \ddot{y} , and \ddot{z} are found from (2.10).

Let M , $\sqrt{-1}N$ and P be the right-hand side terms in (2.6). They are of types \underline{a} , \underline{b} and \underline{c} , respectively. Assume x , y , z have been obtained to order $n-1$. Then M , N and P are known to order n . Some terms of the left-hand side of (2.6) are also known. For instance, if $x = x_1 + x_2 + \cdots + x_{n-1}$, where x_j denotes the terms of order j and, in a similar way, $d = d_0 + d_1 + \cdots + d_{\bar{n}}$, $f = f_0 + f_1 + \cdots + f_{\bar{n}}$, where $\bar{n} = n-2$ if n is even and $\bar{n} = n-3$, if n is odd, then in \dot{x} there are terms already known at order n as

$$\frac{\partial x_{n-2}}{\partial \theta_1} d_2 + \frac{\partial x_{n-2}}{\partial \theta_2} f_2 + \frac{\partial x_{n-4}}{\partial \theta_1} d_4 + \frac{\partial x_{n-4}}{\partial \theta_2} f_4 + \cdots.$$

Transferring those terms of order n , already known when x, y, z are available at order $n - 1$, we obtain new right-hand side terms, whose components at order n are denoted by $\overline{M}_n, \overline{N}_n$ and \overline{P}_n , and we obtain

$$\begin{aligned}
 & \frac{\partial^2 x_n}{\partial \theta_1^2} d_0^2 + \frac{\partial^2 x_n}{\partial \theta_1 \partial \theta_2} 2d_0 f_0 + \frac{\partial^2 x_n}{\partial \theta_2^2} f_0^2 - 2 \left(\frac{\partial y_n}{\partial \theta_1} d_0 + \frac{\partial y_n}{\partial \theta_2} f_0 \right) \\
 & \quad - (1 + 2c_2)x_n + \delta \left(\frac{\partial^2 x_1}{\partial \theta_1^2} 2d_0 d_{n-1} - 2 \frac{\partial y_1}{\partial \theta_1} d_{n-1} \right) = \overline{M}_n, \\
 & \frac{\partial^2 y_n}{\partial \theta_1^2} d_0^2 + \frac{\partial^2 y_n}{\partial \theta_1 \partial \theta_2} 2d_0 f_0 + \frac{\partial^2 y_n}{\partial \theta_2^2} f_0^2 + 2 \left(\frac{\partial x_n}{\partial \theta_1} d_0 + \frac{\partial x_n}{\partial \theta_2} f_0 \right) \\
 & \quad + (c_2 - 1)y_n + \delta \left(\frac{\partial^2 y_1}{\partial \theta_1^2} 2d_0 d_{n-1} + 2 \frac{\partial x_1}{\partial \theta_1} d_{n-1} \right) = \overline{N}_n \sqrt{-1}, \\
 & \frac{\partial^2 z_n}{\partial \theta_1^2} d_0^2 + \frac{\partial^2 z_n}{\partial \theta_1 \partial \theta_2} 2d_0 f_0 + \frac{\partial^2 z_n}{\partial \theta_2^2} f_0^2 \\
 & \quad + c_2 z_n + \delta \left(\frac{\partial^2 z_1}{\partial \theta_2^2} 2f_0 f_{n-1} \right) = \overline{P}_n.
 \end{aligned} \tag{2.12}$$

Here $\delta = 0$ if n is even and $\delta = 1$ if n is odd.

From (2.12) and taking into account the properties 1) to 5) one obtains x_n, y_n, z_n and, if n is odd, also d_{n-1} and f_{n-1} . In the determination of x_n, y_n, z_n one should impose the conditions $x_{i,j,1,0} = 0$ if $(i, j) \neq (1, 0)$ and $z_{i,j,0,1} = 0$ if $(i, j) \neq (0, 1)$ (see property 4). This is what allows to obtain d_{n-1} and f_{n-1} for n odd. More concretely, (2.12) can be written in components as

$$\begin{aligned}
 & -x_{i,j,k,m}(kd_{0,0} + mf_{0,0})^2 + \delta_{k,1}\delta_{m,0}x_{1,0,1,0}(-2d_{0,0}d_{i-1,j}) \\
 & \quad + 2y_{i,j,k,m}(kd_{0,0} + mf_{0,0}) + \delta_{k,1}\delta_{m,0}2y_{1,0,1,0}d_{i-1,j} \\
 & \quad - (1 + 2c_2)x_{i,j,k,m} = \overline{M}_{i,j,k,m}, \tag{2.13}
 \end{aligned}$$

$$\begin{aligned}
 & -y_{i,j,k,m}(kd_{0,0} + mf_{0,0})^2 + \delta_{k,1}\delta_{m,0}y_{1,0,1,0}(-2d_{0,0}d_{i-1,j}) \\
 & \quad + 2x_{i,j,k,m}(kd_{0,0} + mf_{0,0}) + \delta_{k,1}\delta_{m,0}2x_{1,0,1,0}d_{i-1,j} \\
 & \quad + (c_2 - 1)y_{i,j,k,m} = \overline{N}_{i,j,k,m}, \tag{2.14}
 \end{aligned}$$

$$\begin{aligned}
 & -z_{i,j,k,m}(kd_{0,0} + mf_{0,0})^2 \\
 & \quad + \delta_{k,0}\delta_{m,1}z_{0,1,0,1}(-2f_{0,0}f_{i,j-1}) + c_2 z_{i,j,k,m} = \overline{P}_{i,j,k,m}, \tag{2.15}
 \end{aligned}$$

where $\delta_{r,s}$ denotes the delta of Kronecker. From (2.13) and (2.14) if $k = 1$ and $m = 0$ we have

$$\begin{aligned}
 & x_{i,j,1,0}(-d_{0,0}^2 - 1 - 2c_2) + 2y_{i,j,1,0}d_{0,0} \\
 & \quad + d_{i-1,j}(-x_{1,0,1,0}2d_{0,0} + 2y_{1,0,1,0}) = \overline{M}_{i,j,1,0}, \\
 & 2x_{i,j,1,0}d_{0,0} + y_{i,j,1,0}(-d_{0,0}^2 + c_2 - 1) \\
 & \quad + d_{i-1,j}(-y_{1,0,1,0}2d_{0,0} + 2x_{1,0,1,0}) = \overline{N}_{i,j,1,0}.
 \end{aligned}$$

This is solved by putting $x_{i,j,1,0} = 0$ (note that $n \geq 2$) and obtaining then $y_{i,j,1,0}$ and $d_{i-1,j}$, the determinant being different from zero. If $(k, m) \neq (1, 0)$ then one

has

$$\begin{aligned} & \left(-(kd_{0,0} + mf_{0,0})^2 - (1 + 2c_2) \right) x_{i,j,k,m} \\ & \quad + 2(kd_{0,0} + mf_{0,0})y_{i,j,k,m} = \overline{M}_{i,j,k,m}, \\ & \left(-(kd_{0,0} + mf_{0,0})^2 + c_2 - 1 \right) y_{i,j,k,m} \\ & \quad + 2(kd_{0,0} + mf_{0,0})x_{i,j,k,m} = \overline{N}_{i,j,k,m}, \end{aligned} \quad (2.16)$$

which can be solved for $x_{i,j,k,m}$ and $y_{i,j,k,m}$, the determinant being always different from zero.

From (2.15) if $k = 0$ and $m = 1$ we obtain

$$z_{i,j,0,1}(-f_{0,0}^2 + c_2) + z_{0,1,0,1}(-2f_{0,0}f_{i,j-1}) = \overline{P}_{i,j,0,1},$$

which is solved for $f_{i,j-1}$ if one puts $z_{i,j,0,1} = 0$ according to property 4). If $(k, m) \neq (0, 1)$ then

$$\left[-(kd_{0,0} + mf_{0,0})^2 + c_2 \right] z_{i,j,k,m} = \overline{P}_{i,j,k,m}, \quad (2.17)$$

and we obtain $z_{i,j,k,m}$ because $c_2 - (kd_{0,0} + mf_{0,0})^2$ is only zero if $k = 0$, $m = 1$.

Had we not introduced d and f we would find the impossibility to obtain $x_{i,j,1,0}$, $y_{i,j,1,0}$ from (2.16) and $z_{i,j,0,1}$ from (2.17) because of zero divisors. The choices $x_{i,j,1,0} = 0$, $z_{i,j,0,1} = 0$ are arbitrary, but they can be considered as a normalization. In this way the parameters α and β can be interpreted as the coefficients of $\cos \theta_1$ and $\cos \theta_2$, respectively, in the final expressions of x and z .

We summarize the steps to follow to obtain x , y , z to order 2, 3, Assume that they are known to order $n - 1$ (and d , f to order $n - 2$ if n is even, to order $n - 3$ if n is odd). Then we compute ρ^2 to order n , T_m to order n up to $m = n$, R_m to order $n - 1$ up to $m = n - 1$. With this we have M_n , N_n and P_n . Next we obtain the required first and second derivatives of x , y , z with respect to θ_1 , θ_2 and the terms d^2 , $2df$, f^2 to orders 2 to $n - 2$ (resp. $n - 1$) if n is even (resp. if n is odd). From this we get \overline{M}_n , \overline{N}_n , \overline{P}_n and we solve for the $x_{i,j,k,m}$, $y_{i,j,k,m}$, $z_{i,j,k,m}$ with $i + j = n$ and, if n is odd, for $d_{i,j}$ and $f_{i,j}$ with $i + j = n - 1$.

2.3.1 A Program to Obtain the Formal Series Solutions

As it has been said before the properties 3) to 5) allow to reduce the amount of terms to store. For x , y , and z the terms are stored in the following order. A term (i, j, k, m) precedes a term $(\bar{i}, \bar{j}, \bar{k}, \bar{m})$ if $i + j < \bar{i} + \bar{j}$. If $i + j = \bar{i} + \bar{j}$, then if $i > \bar{i}$. If also $i = \bar{i}$ then if $k > \bar{k}$. Finally, if also $k = \bar{k}$, if $m > \bar{m}$. The following instructions allow us to run over all the terms of x , y or z up to order n (note that we do not save the space corresponding to the known fact that $x_{i,j,1,0} = 0$ if $(i, j) \neq (1, 0)$ and that $z_{i,j,0,1} = 0$ if $(i, j) \neq (0, 1)$). The parameter IS takes the value 0 for x and y , and the value 1 for z . The index L is the current index of the term:

```

L = 0
DO 1 NA = 1, N
NAM = NA - IS
NA1 = MOD (NAM, 2)
DO 2 I = NAM, NA1, -2
J = NA - I
DO 3 K = I, NA1, -2
JF = -J
IF (K. EQ. 0) JF = 0
DO 4 M = J, JF, -2
L = L + 1
.....
4 CONTINUE
3 CONTINUE
2 CONTINUE
1 CONTINUE

```

The location where the terms of indices (i, j, k, m) should be stored is computed as $\phi(i, j) + ((i - k)(j + 1) + j + 2 - m)/2$, where ϕ is a function which depends on i and j . It is computed initially and stored in a table. Of course, the expression of ϕ is different for x and for z (for y is the same as for x) but each part takes only a half table. The number of terms for z up to order n is slightly less than for x and y , but we do not save this space. For x and y the total number of terms up to order n , n odd, is given by $(j^4 + 7j^3 + 20j^2 + 26j + 6)/6$, where $j = (n - 1)/2$. For $n = 15, 25$ or 35 this takes the respective values 995, 6005 or 20690.

The functions d, f, d^2, df, f^2 are stored with the element (i, j) at the place $((i + j)(i + j - 2) + 4j + 8)/8$. If n is odd we only require room for them up to order $n - 1$ and each one takes $(j + 1)(j + 2)/2$ positions, where again $j = (n - 1)/2$.

Faced to the alternative more CPU time or more space in memory we have selected the second one (see later). Hence the functions T_m, R_m are kept in memory for all the required values of m . It is possible to keep just two of them for the recurrence. Then T_m, R_m are stored in $T(*, m), R(*, m)$, respectively, where $*$ ranges from 1 to the dimension of x for order n . The maximum value of m is n for T and $n - 1$ for R . We list the routines used to obtain x, y, z, d, f and we note that at order n only terms of degree n (or $n - 1$ for d, f , etc.) are computed, because all the terms of lower degree, required for the intermediate computations, are kept in memory.

Subroutine TAULA Computes the auxiliary function ϕ up to order n .

Subroutine GAMCES Gives $\gamma, c_j, j = 1, 2, \dots, n + 1, \omega_1, \omega_2, \bar{k}$ from μ and the index (1 or 2) of L_i .

Subroutine R02CN Computes the terms of a given order of ρ^2 from x, y, z .

Subroutine PRODN Produces the terms of a given order of the product of two

functions.

Subroutine SUMN Similar to PRODN but for the sum. It also allows to multiply the input functions by a number.

Subroutine TRN Computes the terms of a given order of T_m , R_{m-1} , $m = 2, \dots$, current order, from x and ρ^2 .

Subroutine GET Copies $T(*, j)$ or $R(*, j)$ to a single vector.

Subroutine GETN Same as GET but just for the terms of a given order.

Subroutine PUTN Puts the terms of order n of a vector on $T(*, j)$ or $R(*, j)$.

Subroutine MNPN Computes the terms of a given order of M , N , P from c_j , $j \geq 3$, T_j , R_{j-1} , $j \geq 2$, and y , z .

Subroutine DERN Gives the terms of order n of the required derivatives of x , y , z with respect θ_1 and θ_2 .

Subroutine DDFN Produces the terms of required order of d^2 , $2df$, f^2 from d , f .

Subroutine MNPBN From M , N , P , the previous derivatives and products d^2 , etc. gives \bar{M} , \bar{N} , \bar{P} .

Subroutine PROLCN Entering a series as x , y , z and another one as d , f , d^2 , etc. gives the product.

Subroutine SOLVEN Solves the terms of order n of x , y and z and, if n is odd, the ones of d , f of degree $n - 1$.

Subroutine OUT Outputs the results.

We refer to the listings of the programs for a more detailed description and also to the Users and Programmers Guide [4].

Finally we make some a priori comments on the running space and CPU time. The programs have been structured in such a way that we can modify a few parameter instructions by setting

$$NG = (j^4 + 7j^3 + 20j^2 + 26j + 6)/6, \quad NG1 = 2j + 1, \quad NG2 = (j^2 + 3j + 2)/2,$$

if the maximal order to run the program is $n = 2j + 1$. One needs $20 + 2n$ vectors of dimension NG and a few ones of dimensions $NG1$, $NG2$. The total space required (in double precision) is $(4j^5 + 56j^4 + 236j^3 + 691j^2 + 881j)/6 + \text{ctant}$. If $n = 35$ this is close to 15.5 Mb, and it is close to 4 Mb if $n = 25$. To keep in mind a simpler expression, the required space up to order n is proportional to $n^4(n + 23)$ for n large. An analysis of the program shows that the CPU time is roughly proportional to n^8 . Had we stored just two consecutive terms of $\{T_m\}$, $\{R_m\}$ the required space would be proportional to n^4 but the computing time would be to n^9 , becoming prohibitive for large orders.

2.3.2 Results and Formal Tests

The program described in the above section, named LISFIL, has been run for several values of n , for L_1 and $\mu = 0.3040357143 \times 10^{-5}$ (the adopted value of the Barycenter

n	9	11	13	15	17	19	21	23	25	35
CPU	0.4	1.4	3.6	9.1	20.9	44.6	91.4	176.4	328.0	4338.7

Table 2.1 CPU time (in seconds), for different values of n .

/ (Sun + Barycenter) mass ratio). The CPU times on an HP 9000 / 835S computer are shown in Table 2.1.

Tables 2.2 and 2.3 give, respectively, the values of $d_{i,j}$, $f_{i,j}$ up to order 21 and of x , y , z up to order 5.

To have a first look to the results Figures 2.1 to 2.4 show the analytic orbits computed to order 35 for the following values of (α, β) : (0.1, 0.05), (0.1, 0.3), (0.05, 0.3), (0, 0.3). In all the plots the (y, z) projection is displayed and the unit of length is γ (roughly 1.5×10^6 km).

The results have been formally tested in the following sense. With the solution available in the form of (2.10) it has been substituted in (2.6). The left- and right-hand sides must be equal. To allow for rounding errors if one term to the left and to the right should be exactly zero (because of the symmetries) we have asked it to be less than 10^{-15} in absolute value. Otherwise it has been required that the absolute value of the difference be less than 10^{-10} times the maximum of the absolute values of the corresponding terms in both sides. Program TESLAN did the check and found that up to order 16 all the elements passed the test. For increasing orders, specially beyond order 30, some 500 terms (including x , y , z) did not passed the test, reaching relative errors of 10^{-4} for a few ones of them. But most of those are due to terms which are rather small.

$d_{i,j}$	$f_{i,j}$	$d_{i,j}$	$f_{i,j}$
0.2086453455276D+01	0.2015210551475D+01	0.1509859603685D+07	-0.4003960409393D+06
-0.1720615798629D+01	0.2227433602201D+00	-0.3084398700348D+05	0.2668896522823D+04
0.2581845049892D-01	-0.1631915163450D+00	0.1485537635945D+03	-0.3616326120071D-04
0.1311254416238D+00	0.2139360037470D+02	-0.8670452284941D+03	0.4546558058958D+11
0.5328997881953D+01	-0.5221779021254D+01	0.1270725435729D+11	-0.1268031002099D+12
-0.2985293861854D+00	-0.3191736342282D-01	-0.2655062348622D+11	0.8096465746173D+11
0.5795735495703D+01	0.5786805804301D+03	0.1348950273657D+11	-0.1980965473829D+11
0.1573285467464D+03	-0.3489189300595D+03	-0.2693230891169D+10	0.2165574122979D+10
-0.4094951681817D+02	0.1202414068604D+02	0.2426390383752D+09	-0.1073138329738D+09
0.6853368228154D+00	-0.1343860290995D-02	-0.9879489518851D+07	0.2150775686649D+07
-0.1340765219537D+02	0.1883713317349D+05	0.1592949183871D+06	-0.1169277155681D+05
0.5139709471321D+04	-0.1926740060050D+05	-0.6481376995677D+03	-0.3095886545754D-04
-0.2915891224396D+04	0.2298536607974D+04	0.8348466194561D+04	0.1958490875656D+13
0.2210923683071D+03	-0.4068836253461D+02	0.5491139931955D+12	-0.6346243374055D+13
-0.2324776780509D+01	0.1889963016498D-02	-0.1382615079745D+13	0.4990901097465D+13
-0.1343928579873D+02	0.6851083943067D+06	0.8837538292374D+12	-0.1589383667217D+13
0.1885553336878D+06	-0.9961508607394D+06	-0.2338607980291D+12	0.2430345868831D+12
-0.1717925036809D+06	0.2303860273217D+06	0.3005263841885D+11	-0.1878048416470D+11
0.2802535526097D+05	-0.1330324946783D+05	-0.1957754838468D+10	0.7109157206391D+09
-0.1154811677834D+04	0.1551582333135D+03	0.6210875802615D+08	-0.1148638810566D+08
0.8743071596302D+01	0.8193752746315D-03	-0.8238976392160D+06	0.5249722876900D+05
0.1769835787653D+03	0.2661783973085D+08	0.2900339414339D+04	-0.9151232763487D-05
0.7373934058941D+07	-0.5042332873656D+08	-0.1704015556269D+05	0.8603177589146D+14
-0.9460878241000D+07	0.1807515142716D+08	0.2418415527902D+14	-0.3177546525092D+15
0.2532029829402D+07	-0.2022817837248D+07	-0.7160813108330D+14	0.2979994444608D+15
-0.2168942262597D+06	0.7374867895792D+05	0.5555722778467D+14	-0.1178237880998D+15
0.5975062228439D+04	-0.6291996739779D+03	-0.1851399736985D+14	0.2350421188856D+14
-0.3520922885183D+02	0.1315381735879D-03	0.3147047168790D+13	-0.2539286156383D+13
-0.4065532400348D+03	0.1082133781426D+10	-0.2912789554730D+12	0.1495201494611D+12
0.3012679588378D+09	-0.2532155414999D+10	0.1469539777359D+11	-0.4540967859281D+10
-0.5052356674707D+09	0.1254011881097D+10	-0.3797951473631D+09	0.6114487265964D+08
0.1938173328512D+09	-0.2197171658229D+09	0.4269034413606D+07	-0.2402675177845D+06
-0.2702232053789D+08	0.1536013372135D+08	-0.1323833340710D+05	-0.1415565469857D-06

Table 2.2 Values of $d_{i,j}$ and $f_{i,j}$ up to order 21. Libration point: L_1 .

$$\mu = 3.04035714299999895\text{E-}06. \gamma = 1.00109047548951804\text{E-}02$$

1 0 1 0	0.5000000000D+0	0.1614634048D+1	0 1 0 1	0.5000000000D+0
2 0 2 0	-0.4529823976D+0	0.2462229375D+0	1 1 1 1	0.1774726533D+0
2 0 0 0	0.2092695580D+1	0.0000000000D+0	1 1 1-1	-0.5584341689D+0
0 2 0 2	0.5541259564D-1	0.3388187150D-1	2 1 2 1	0.2030396443D+0
0 2 0 0	0.2482976702D+0	0.0000000000D+0	2 1 2-1	0.6082827416D+1
3 0 3 0	-0.3969100975D+0	0.4428503881D+0	2 1 0 1	0.0000000000D+0
3 0 1 0	0.0000000000D+0	-0.1422540573D+1	0 3 0 3	-0.9763610395D-2
1 2 1 2	0.4193889016D-1	-0.1041440389D-1	0 3 0 1	0.0000000000D+0
1 2 1 0	0.0000000000D+0	-0.1437765435D+0	3 1 3 1	0.2437559135D+0
1 2 1-2	-0.7499973544D+0	0.2420983547D+1	3 1 3-1	0.2385997972D+1
4 0 4 0	-0.5950722377D+0	0.5373563863D+0	3 1 1 1	0.7652676977D-1
4 0 2 0	0.8702523463D+0	-0.9602066054D+0	3 1 1-1	-0.7673237490D+1
4 0 0 0	-0.4013683102D+1	0.0000000000D+0	1 3 1 3	-0.1672719566D-1
2 2 2 2	0.8609358834D-1	-0.1900491048D-1	1 3 1 1	-0.3840223284D-2
2 2 2 0	0.1424392122D+1	0.8353734089D+0	1 3 1-1	0.7632863955D+0
2 2 2-2	0.2045980254D+0	0.6030548917D-1	1 3 1-3	-0.2568961341D+0
2 2 0 2	0.1380923671D+1	-0.5762991637D+0	4 1 4 1	0.3477722470D+0
2 2 0 0	0.2577287869D+0	0.0000000000D+0	4 1 4-1	0.2758896900D+1
0 4 0 4	-0.2662023471D-2	-0.1040300651D-2	4 1 2 1	-0.6030868796D-1
0 4 0 2	0.3621865083D-2	0.6172566421D-2	4 1 2-1	0.1663319440D+3
0 4 0 0	0.2920336247D-2	0.0000000000D+0	4 1 0 1	0.0000000000D+0
5 0 5 0	-0.8436465974D+0	0.8169824737D+0	2 3 2 3	-0.3543999334D-1
5 0 3 0	0.1288359895D+1	-0.1551690740D+1	2 3 2 1	-0.3701752729D+0
5 0 1 0	0.0000000000D+0	0.1933414620D+1	2 3 2-1	-0.2362736904D+2
3 2 3 2	0.1379491354D+0	-0.6480700010D-1	2 3 2-3	0.7833794866D+1
3 2 3 0	0.1082618822D+1	-0.2259367366D+0	2 3 0 3	-0.5959980280D+0
3 2 3-2	0.1000012891D+2	0.3280956762D+2	2 3 0 1	0.0000000000D+0
3 2 1 2	0.1695735617D+1	-0.1778432650D+1	0 5 0 5	0.6872980412D-3
3 2 1 0	0.0000000000D+0	-0.1397330649D+0	0 5 0 3	-0.5545734449D-3
3 2 1-2	-0.3226760332D+2	0.1020897376D+3	0 5 0 1	0.0000000000D+0
1 4 1 4	-0.5443907814D-2	0.2802932442D-3	5 1 5 1	0.5200781298D+0
1 4 1 2	0.1946275268D-2	0.8806522073D-3	5 1 5-1	0.3337336494D+1
1 4 1 0	0.0000000000D+0	-0.6053359116D-1	5 1 3 1	-0.2802699258D+0
1 4 1-2	0.1799803982D+1	-0.5977698999D+1	5 1 3-1	0.6000805955D+2
1 4 1-4	-0.5882638989D-1	-0.1252672535D-1	5 1 1 1	0.2548515339D+1
6 0 6 0	-0.1316113331D+1	0.1252252243D+1	5 1 1-1	-0.1965457895D+3
6 0 4 0	0.2393231791D+1	-0.2457065108D+1	3 3 3 3	-0.6733054547D-1
6 0 2 0	-0.5129057343D+0	0.8904712899D+0	3 3 3 1	-0.6267328906D+0
6 0 0 0	0.2710980997D+1	0.0000000000D+0	3 3 3-1	-0.4915761419D+1
4 2 4 2	0.2548959901D+0	-0.1346515671D+0	3 3 3-3	-0.9586916266D+1
4 2 4 0	0.2173912950D+1	-0.3961207334D+0	3 3 1 3	-0.1047186013D+1
4 2 4-2	-0.1050812422D+2	0.1656217046D+2	3 3 1 1	-0.3841453332D+1
4 2 2 2	0.3298146323D+1	-0.2833728342D+1	3 3 1-1	0.6267921875D+2
4 2 2 0	0.3748270746D+2	0.2654917935D+2	3 3 1-3	-0.8920296866D+1
4 2 2-2	0.5167313415D+1	0.2106326578D+0	1 5 1 5	0.2072873152D-2
4 2 0 2	0.5555843733D+2	-0.2743863952D+2	1 5 1 3	-0.1887639034D-2
4 2 0 0	0.5286381312D+2	0.0000000000D+0	1 5 1 1	-0.2640134141D-1
2 4 2 4	-0.1467089328D-1	0.1947343820D-2	1 5 1-1	-0.1876434349D+1
2 4 2 2	-0.1277954836D+0	-0.5350069831D-1	1 5 1-3	0.6376782691D+0
2 4 2 0	-0.5070532300D+1	-0.3219418900D+1	1 5 1-5	0.2286440887D-1
2 4 2-2	-0.5707159358D+0	0.2299448683D-1		
2 4 2-4	0.7437063971D+0	-0.1468237810D+1		
2 4 0 4	-0.2490486627D+0	0.4244008555D-1		
2 4 0 2	-0.3440638315D+1	0.1773083619D+1		
2 4 0 0	0.2723350364D+1	0.0000000000D+0		
0 6 0 6	0.2220978415D-3	0.7091420173D-4		
0 6 0 4	-0.3597237702D-3	-0.3004542108D-3		
0 6 0 2	-0.6930115610D-3	0.6826458318D-3		
0 6 0 0	-0.4968213471D-2	0.0000000000D+0		

Table 2.3 Indices i, j, k, m and values of $x_{i,j,k,m}$, $y_{i,j,k,m}$ and $z_{i,j,k,m}$ for the Lissajous orbit of order 5. Equilibrium point: L_1 .

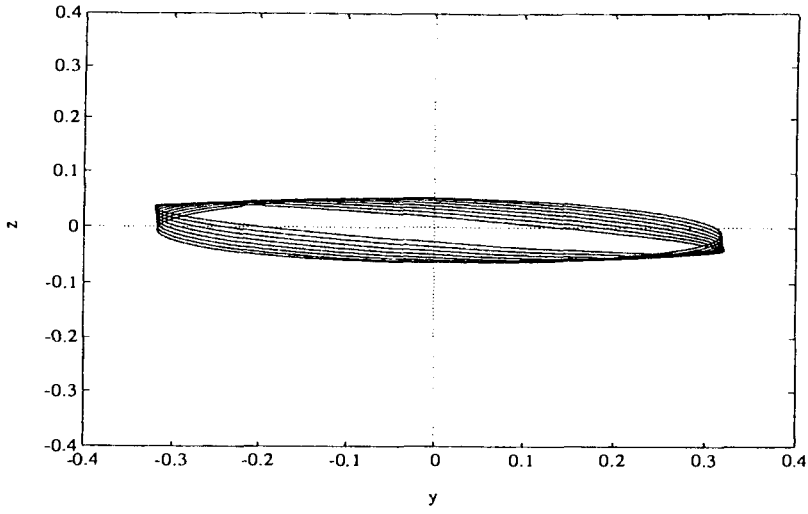


Fig. 2.1 Lissajous orbit for $\alpha = 0.1$, $\beta = 0.05$, $\varphi_1 = \varphi_2 = 0$. Final time $t_f = 25$.

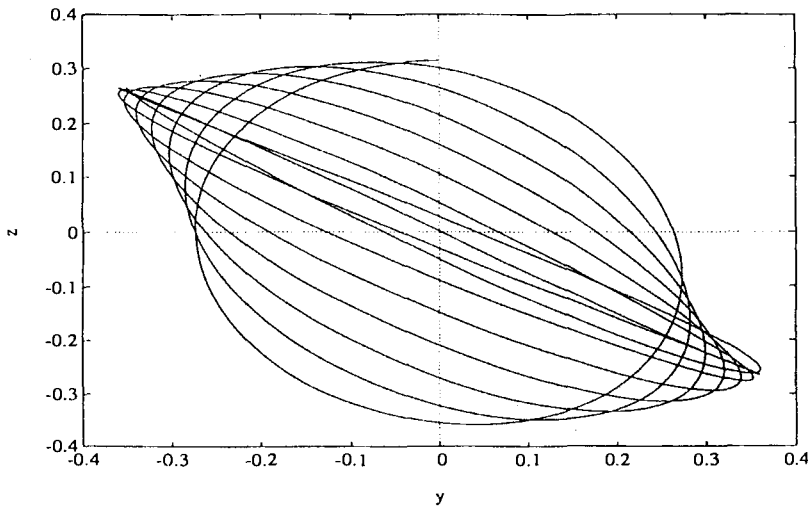


Fig. 2.2 Lissajous orbit for $\alpha = 0.1$, $\beta = 0.3$, $\varphi_1 = \varphi_2 = 0$. Final time $t_f = 25$.

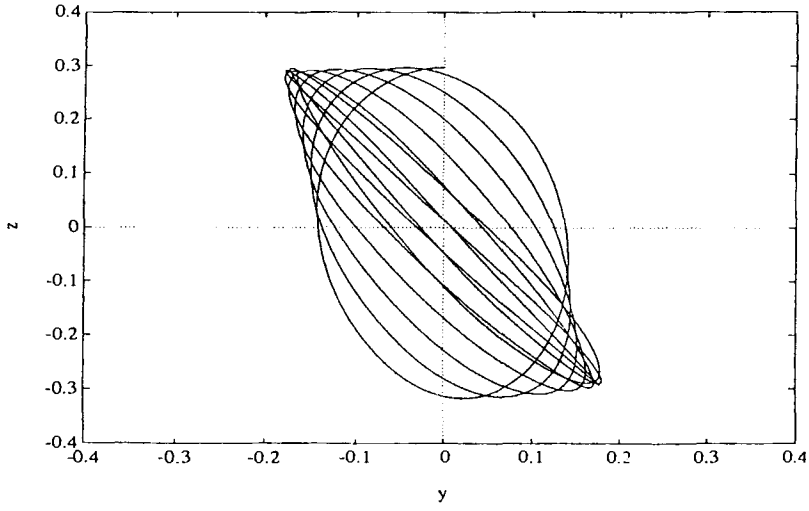


Fig. 2.3 Lissajous orbit for $\alpha = 0.05$, $\beta = 0.3$, $\varphi_1 = \varphi_2 = 0$. Final time $t_f = 25$.

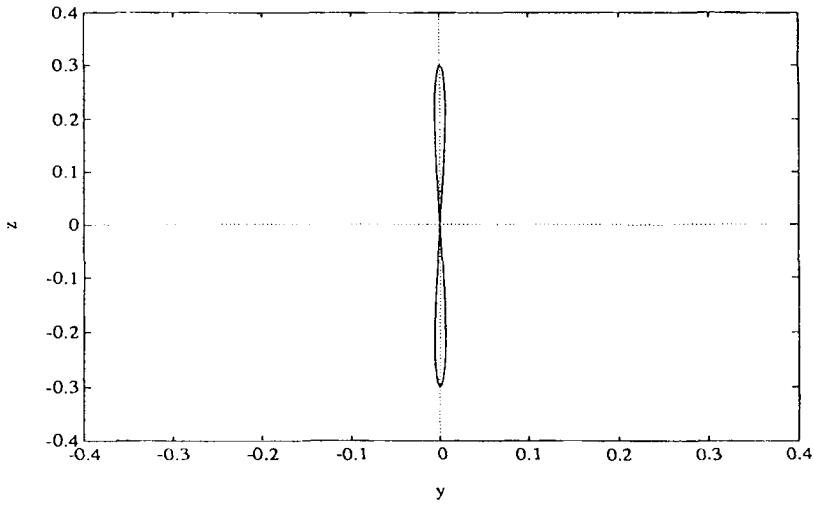


Fig. 2.4 Vertical periodic orbit $\alpha = 0$, $\beta = 0.3$. $\varphi_1 = \varphi_2 = 0$ Period = 3.14117812.

2.3.3 Numerical Tests

The analytic solutions of (2.1) have been checked against numerical solutions by program TESLNU as follows. Given $\alpha, \beta, \phi_1, \phi_2$ and some order n we obtain the initial conditions for $X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$ at $t = 0$. Then the numerical integration has been started and the distance between the numerical solution and the analytic solution is computed (in the phase space) at regular intervals of 0.01 time units (2π time units = 1 year). This test has been done for orders 15, 25 and 35, for $\alpha = 0.05$ and 0.10, for $\beta = 0.05(0.05)0.30$ and $\phi_1 = j_1\pi/4, j_1 = 0, \dots, 7, \phi_2 = j_2\pi/4, j_2 = 0, \dots, 3$, (the values for $\phi_2 \geq \pi$ are not necessary due to the symmetry). A summary of results is shown in Table 2.4.

From the test we can extract the following conclusions:

- (1) Going from order 15 to order 25 the results improve by a factor of the order of 1000.
- (2) Going from order 25 to order 35 the results do not improve for $\alpha = 0.05, \beta \leq 0.15$. This is due to the fact that the errors of the theory are of the order of magnitude of the rounding error. Otherwise they improve by a factor of 100.
- (3) For order 35 the errors in the phase space, expressed as a distance (1 unit = 1 AU) after 6.28 time units are less than 2km if $\alpha \leq 0.05, \beta \leq 0.25$. For $\alpha = 0.1$ or $\beta = 0.3$ can reach 30 000 km, but only 14 km after 3.14 time units.
- (4) The quotients of the distances at $t = 6.28$ and $t = 3.14$ (and also at $t = 4.71$ and $t = 1.57$) are between 2 000 and 3 000 in all cases, at most. This reflects the inherent instability of those orbits due to the instability of L_1 .

Maximum error after 1 revolution and indices of initial conditions		
$\beta = 0.05$	$0.23778290D - 08$	0 3
$\beta = 0.10$	$0.19964676D - 08$	0 3
$\beta = 0.15$	$0.67030706D - 08$	0 0
$\beta = 0.20$	$0.70044099D - 07$	1 2
$\beta = 0.25$	$0.23740954D - 05$	1 0
$\beta = 0.30$	$0.22665829D - 04$	1 2
Maximum error after 2 revolutions and indices of initial conditions		
$\beta = 0.05$	$0.64756188D - 05$	0 3
$\beta = 0.10$	$0.53753392D - 05$	0 3
$\beta = 0.15$	$0.17692378D - 04$	0 0
$\beta = 0.20$	$0.17937577D - 03$	1 2
$\beta = 0.25$	$0.52602829D - 02$	1 0
$\beta = 0.30$	$0.38554911D - 01$	0 3
Maximum quotient of errors in 3/2 and 1/2 revolutions		
$\beta = 0.05$	$0.38043003D + 04$	5 1
$\beta = 0.10$	$0.29739554D + 04$	5 1
$\beta = 0.15$	$0.33117953D + 04$	1 3
$\beta = 0.20$	$0.28057723D + 04$	2 1
$\beta = 0.25$	$0.33228296D + 04$	7 0
$\beta = 0.30$	$0.27354571D + 04$	7 2
Maximum quotient of errors in 3/2 and 1 revolutions		
$\beta = 0.05$	$0.27551052D + 04$	6 1
$\beta = 0.10$	$0.27247626D + 04$	6 1
$\beta = 0.15$	$0.26907616D + 04$	6 3
$\beta = 0.20$	$0.27757272D + 04$	5 3
$\beta = 0.25$	$0.28078203D + 04$	7 2
$\beta = 0.30$	$0.45627723D + 04$	7 2

Table 2.4.1. Differences between the analytic and the numerical solutions. Order 15. Only results for $\alpha = 0.05$. Breakdown for $\alpha = 0.10$.

Maximum error after 1 revolution and indices of initial conditions				
$\beta = 0.05$	$0.47901007D - 11$	1 3	$0.21175892D - 05$	7 0
$\beta = 0.10$	$0.41486226D - 11$	3 3	$0.12020701D - 05$	0 3
$\beta = 0.15$	$0.47249882D - 11$	7 0	$0.11568056D - 05$	0 3
$\beta = 0.20$	$0.12913489D - 10$	1 1	$0.13681264D - 05$	0 3
$\beta = 0.25$	$0.14919073D - 08$	5 0	$0.25652921D - 05$	0 3
$\beta = 0.30$	$0.84202000D - 07$	4 3	$0.71966246D - 05$	0 3
Maximum error after 2 revolutions and indices of initial conditions				
$\beta = 0.05$	$0.12945928D - 07$	1 3	$0.46810993D - 02$	7 0
$\beta = 0.10$	$0.11159943D - 07$	3 3	$0.27150046D - 02$	0 3
$\beta = 0.15$	$0.12602806D - 07$	7 0	$0.29055378D - 02$	0 3
$\beta = 0.20$	$0.32950321D - 07$	1 1	$0.29274034D - 02$	0 3
$\beta = 0.25$	$0.37275552D - 05$	5 0	$0.64842637D - 02$	0 3
$\beta = 0.30$	$0.20192071D - 03$	4 3	$0.11579621D - 01$	0 3
Maximum quotient of errors in 3/2 and 1/2 revolutions				
$\beta = 0.05$	$0.33993263D + 04$	1 0	$0.28129668D + 04$	5 3
$\beta = 0.10$	$0.30762918D + 04$	7 2	$0.27122272D + 04$	6 3
$\beta = 0.15$	$0.27637071D + 04$	3 1	$0.34955116D + 04$	5 1
$\beta = 0.20$	$0.30855018D + 04$	5 2	$0.31539157D + 04$	4 1
$\beta = 0.25$	$0.29849801D + 04$	5 2	$0.27523700D + 04$	1 1
$\beta = 0.30$	$0.27294821D + 04$	0 1	$0.26448217D + 04$	5 1
Maximum quotient of errors in 3/2 and 1 revolutions				
$\beta = 0.05$	$0.27532599D + 04$	6 0	$0.27200436D + 04$	7 2
$\beta = 0.10$	$0.27351259D + 04$	6 1	$0.25090266D + 04$	4 1
$\beta = 0.15$	$0.26742006D + 04$	6 2	$0.25187494D + 04$	7 3
$\beta = 0.20$	$0.26013352D + 04$	6 2	$0.23711618D + 04$	7 2
$\beta = 0.25$	$0.25141232D + 04$	6 2	$0.25707812D + 04$	7 3
$\beta = 0.30$	$0.24181732D + 04$	6 3	$0.22874725D + 04$	7 2

Table 2.4.2. Differences between the analytic and the numerical solutions. Order 25. Left columns for $\alpha = 0.05$, right ones for $\alpha = 0.10$.

Maximum error after 1 revolution and indices of initial conditions				
$\beta = 0.05$	$0.47900960D - 11$	1 3	$0.46814116D - 07$	0 3
$\beta = 0.10$	$0.41486245D - 11$	3 3	$0.11072099D - 07$	7 0
$\beta = 0.15$	$0.47249264D - 11$	7 0	$0.10598980D - 07$	0 3
$\beta = 0.20$	$0.42194950D - 11$	4 0	$0.90004112D - 08$	0 3
$\beta = 0.25$	$0.46330278D - 11$	4 3	$0.20084169D - 07$	7 0
$\beta = 0.30$	$0.66745828D - 09$	5 0	$0.93927219D - 07$	0 3
Maximum error after 2 revolutions and indices of initial conditions				
$\beta = 0.05$	$0.12945928D - 07$	1 3	$0.11475146D - 03$	7 0
$\beta = 0.10$	$0.11159943D - 07$	3 3	$0.27295771D - 04$	7 0
$\beta = 0.15$	$0.12602722D - 07$	7 0	$0.24941371D - 04$	0 3
$\beta = 0.20$	$0.10884827D - 07$	4 0	$0.20588991D - 04$	0 3
$\beta = 0.25$	$0.11542052D - 07$	4 3	$0.45521867D - 04$	7 0
$\beta = 0.30$	$0.15987905D - 05$	5 0	$0.19711410D - 03$	0 3
Maximum quotient of errors in 3/2 and 1/2 revolutions				
$\beta = 0.05$	$0.34005577D + 04$	1 0	$0.28177878D + 04$	5 3
$\beta = 0.10$	$0.37818623D + 04$	7 2	$0.27661023D + 04$	0 0
$\beta = 0.15$	$0.27296255D + 04$	5 3	$0.33692239D + 04$	5 1
$\beta = 0.20$	$0.30302109D + 04$	1 2	$0.27415258D + 04$	5 1
$\beta = 0.25$	$0.33081692D + 04$	3 1	$0.25842368D + 04$	5 1
$\beta = 0.30$	$0.30195014D + 04$	1 2	$0.26235798D + 04$	1 1
Maximum quotient of errors in 3/2 and 1 revolutions				
$\beta = 0.05$	$0.27532639D + 04$	6 0	$0.26178837D + 04$	2 2
$\beta = 0.10$	$0.27350613D + 04$	6 1	$0.26142558D + 04$	6 2
$\beta = 0.15$	$0.26747531D + 04$	6 2	$0.24174800D + 04$	7 3
$\beta = 0.20$	$0.27366497D + 04$	6 3	$0.23518036D + 04$	7 3
$\beta = 0.25$	$0.25156315D + 04$	6 2	$0.22736985D + 04$	7 3
$\beta = 0.30$	$0.24098099D + 04$	6 3	$0.21772284D + 04$	6 2

Table 2.4.3. Differences between the analytic and the numerical solutions. Order 35. Left columns for $\alpha = 0.05$, right ones for $\alpha = 0.10$.

2.4 On the Convergence of the Series

If we consider the Hamiltonian formalism, the second order terms, in suitable variables, are of the form

$$H = \lambda x_1 y_1 + \frac{1}{2} \omega_1 (x_2^2 + y_2^2) + \frac{1}{2} \omega_2 (x_3^2 + y_3^2),$$

where $\lambda x_1 y_1$ is the responsible of the instability and the other two are two harmonic oscillators. Higher order terms couple the three modes. If we skip the instability (see the next section) we have a two degrees of freedom Hamiltonian whose lower order terms are a couple of harmonic oscillators. Then we shall have, close to the origin, a big set of KAM tori*. But, between those tori, there are very narrow stochastic zones (because the resonances are dense). Hence we shall have divergence everywhere (except at the origin, but this is not very useful).

However, as said in the Introduction, small divisors shall show up only at high orders (except the one due to the 1 to 1 resonance) because, at the origin ω_1/ω_2 is close to 29/28. The high order resonances have a very small stochastic zone and the effect is only seen after a big time interval.

Looking for practical applications with the analytic results up to order 35 we did the following test: Let $\alpha = \sigma \cos \epsilon$, $\beta = \sigma \sin \epsilon$. By substituting in d and f we found

$$d = \sum_{j=0, j \text{ even}}^{34} c_{1,j} \sigma^j, \quad f = \sum_{j=0, j \text{ even}}^{34} c_{2,j} \sigma^j, \quad c_{i,j} = c_{i,j}(\epsilon).$$

Let

$$r(\epsilon) = \frac{0.75}{\sup_{i,j} |c_{i,j}|^{1/j}}.$$

This gives a “practical” radius of convergence. The factor 0.75 ensures that, if $r(\epsilon)$ were the true radius of convergence, the remainder would be less than 10^{-4} .

Figure 2.5 shows the “domain of practical convergence” with this criterion when ϵ ranges from 0 to $\pi/2$.

One can also check that in this domain d decreases and f increases if α increases or if β decreases, provided β is not too large. Otherwise both d and f decrease when α increases or β decreases. Rather than to give the plots of d and f in terms of α and β we computed more useful things. Let $\Delta = d - f$ and $q = d/f$. The lines $q = \text{ctant} \in \mathbb{Q}$ give resonances, and the curves $\Delta = \text{ctant}$ give constant difference of frequencies, to be used in the discussion. Figure 2.5 also displays curves of constant

*This refers to the invariant tori which persist when we pass from an integrable Hamiltonian system to a small perturbation of it. The existence of these tori is ensured by the celebrated Kolmogorov-Arnold-Moser Theorem. The measure of the set of points, in the phase space, which were in a torus for the integrable system and are not in an invariant torus (but in some “chaotic” region) when the perturbation is added, goes to zero when the size of the perturbation does.

q (dotted), for q ranging from 809/800 to 839/800, and curves $\Delta = \text{const}$, for Δ ranging from 0.025 to 0.095 (solid lines).

Assume that the real problem is the RTBP and we use the formal solution as nominal orbit. Due to the instability one should do manoeuvres from time to time. Hence we can allow for some small error in the nominal orbit. This one will be corrected when the instability is. Hence we can take some values of α and β . Search for ϕ_1 and ϕ_2 such that after 3 months the difference between the analytic and numerical solution is maximal (in the phase space). We decide that if this difference is less than 10^{-7} the values of (α, β) are admissible. The boundary of the region of admissible parameters is also displayed in Figure 2.5, and it gives another "practical use of the method" zone. It is slightly smaller than the region of "practical" convergence found before for d and f .

As a conclusion, despite the nonconvergent character of the series they can be used, for some practical purposes, in some big region of the α, β parameter range. They produce approximations to invariant tori: the orbits on them are quasi-periodic if $d/f \notin \mathbb{Q}$ and of long period if $d/f \in \mathbb{Q}$. They can be seen as nonlinear Lissajous orbits.

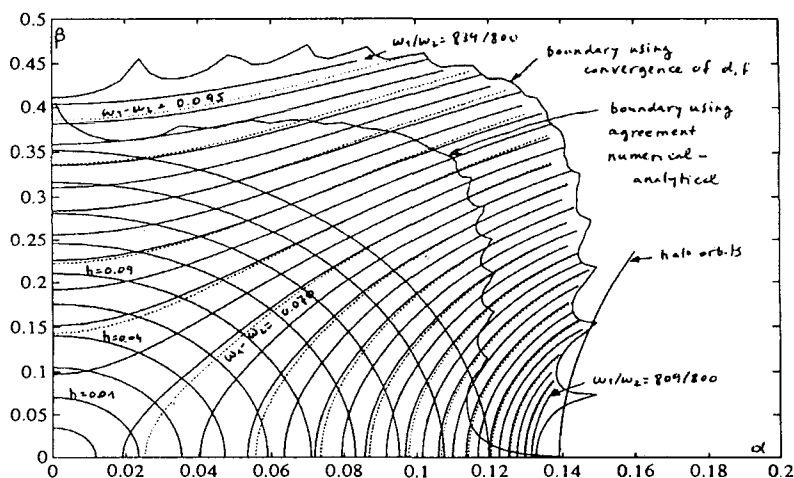


Fig. 2.5

Finally we would like to point out that the expressions found for the Lissajous orbits remain valid for $\beta = 0$ (Lyapunov periodic orbits) and for $\alpha = 0$ (vertical periodic orbits). In these cases no problems due to small divisors show up and the restricted series are convergent. Working at order 35, for $\beta = 0$ values of α up to 0.35 give very good approximations. If $\alpha = 0$, β can reach values up to 1.13.

2.5 Towards a Description of the Neighbourhood of L_1

In this section we sketch the ideas to be developed in the subsequent ones. First of all we think about skipping the unpleasant unstable terms of the Hamiltonian. This is accomplished by the reduction to the central manifold W^c of dimension 4. We describe how to carry out the computations. After performing a linear change of variables going from x, y, p_x, p_y to x_1, x_2, y_1, y_2 and putting $x_3 = z, y_3 = p_z$, we have that the Hamiltonian K given by 2.7 can be written as

$$M = \lambda x_1 y_1 + \frac{1}{2} \omega_1 (x_2^2 + y_2^2) + \frac{1}{2} \omega_2 (x_3^2 + y_3^2) + \sum_{j \geq 3} M_j(x_1, x_2, x_3, y_1, y_2, y_3).$$

Then we pass to an intermediate normal form just trying to kill in M all the terms of the form $x_1^{k_1} y_1^{l_1}$ with $k_1 \neq l_1$. This is possible because there are not problems of small divisors. The denominators appearing in the generating function (using some complexification to reduce the amount of computation) are of the form

$$(k_1 - l_1)\lambda + \sqrt{-1}((k_2 - l_2)\omega_1 + (k_3 - l_3)\omega_2),$$

with modulus bounded from below by λ . Let \widetilde{M} be this intermediate Hamiltonian. By rearranging terms it can be written as

$$\widetilde{M} = M^0(x_2, x_3, y_2, y_3) + \sum_{j \geq 1} (x_1 y_1)^j M^j(x_2, x_3, y_2, y_3).$$

It is immediately seen that $I_1 = x_1 y_1$ is a first integral. The reduction to the center manifold is obtained by setting $I_1 = 0$. So it remains a two degrees of freedom Hamiltonian, M^0 , and furthermore

$$M^0 = \frac{1}{2} \omega_1 (x_2^2 + y_2^2) + \frac{1}{2} \omega_2 (x_3^2 + y_3^2) + \sum_{j \geq 3} M_j^0(x_2, x_3, y_2, y_3). \quad (2.18)$$

We point out that if in (2.18) we put $x_3 = y_3 = 0$ we have an invariant set under the flow associated to M^0 . This is now a 1 degree of freedom Hamiltonian. The orbits are simply the Lyapunov periodic orbits around L_1 .

The levels of energy of M^0 are also represented on Figure 2.5. Each level corresponds to an S^3 sphere. When $\beta = 0$ we obtain a Lyapunov orbit. When $\alpha = 0$ an orbit as the one displayed in Figure 2.4. For $\alpha \neq 0$ and $\beta \neq 0$ we have either tori or stochastic zones. The orbits in Figures 2.1, 2.2 and 2.3 show (approximations of) typical orbits in the tori.

Keeping $M^0 = h$ fixed we can use $z = x_3 = 0$ with $\dot{z} = y_3$ as surface of section. This section, Σ , is not global because it fully contains the corresponding Lyapunov orbit. But we can think that all the points of this orbit (in the boundary of the section) are identified. So we should obtain an S^2 sphere (the classical Hopf fibration) that can be seen as a space of “osculating orbits”. For some value of h , h_H say, the level $H^0 = h$ contains the Lyapunov orbit bifurcating to halo orbits.

From this value on the section Σ contains two fixed points associated to the two (symmetrical) halo orbits. It is known that, at least for moderate values of h , they are of elliptic type. At least for values of h not too far from h_H we have in S^2 four fixed points. Two of them are the Lyapunov orbit (hyperbolic) and the almost vertical periodic orbit as in Figure 2.4 (elliptic). The other two are the halo orbits (elliptic). So the sum of the indices (1 for elliptic and -1 for hyperbolic) agrees with the Euler-Poincaré characteristic of S^2 .

2.6 Discussion on the Use of Lissajous Orbits

The Lissajous orbits should avoid them, for practical purposes, a neighborhood of the Earth–Sun system line (to prevent an spacecraft like SOHO to stay for some time in a region where the radio communications are distorted by the noise coming from the Sun). Let l be the minimum admissible angle between the Earth–Sun and Earth–spacecraft lines. For technical reasons l should be, at least, 3 degrees.

On the other hand orbits making small excursions on the y and z directions are desirable to minimize the cost of the antenna. The admissible halo orbits give excursions of the order of 26° in the y direction to each side of the Earth–Sun line. Hence we can think about the use of Lissajous orbits. However those ones have the problem that they shall pass in front of the solar disk due to the increasing delay between the phases. The standing problems are: How long it takes to pass too close to the Earth–Sun line and, if they do this, which kind of manoeuvre one should do to shift the motion to the admissible region. The answer to those questions is given in what follows for the RTBP.

To avoid too frequent passages near the Sun direction one should require that the two basic frequencies of the Lissajous orbits, ω_1 and ω_2 , be as close as possible. In other words, the difference $\Delta = d - f = \Delta(\alpha, \beta)$ should be small. Looking at Figure 2.5 this is seen to be not compatible with the requirement that α and β be small. A more detailed discussion follows.

Suppose we start with initial phases $\phi_1 = \phi_2 = 0$ at $t = 0$, with amplitudes α and β . Then, after some time, T , we have that the relation

$$\frac{(x^2 + y^2)^{1/2}}{1 - x} = \tan l$$

holds for the first time, where x, y, z denote the position of the spacecraft in normalized coordinates. Furthermore we can also look, in the time interval $[0, T]$, for the maximum values of

$$a_y = \tan^{-1} \frac{|y|}{1 - x} \quad \text{and} \quad a_z = \tan^{-1} \frac{|z|}{1 - x}.$$

Keeping $l = 3^\circ$ we scanned several couples (α, β) . We keep just two such couples:

- (1) If $\alpha = 0.0484$ and $\beta = 0.14874$ then $a_y \simeq a_z \simeq 9^\circ$ and $T = 14.94$ units.
(We recall that 2π units = 1 year).
- (2) If $\alpha = 0.0386$ and $\beta = 0.1228$ then $a_y \simeq a_z \simeq 7^\circ.25$ and $T = 14.87$ units.

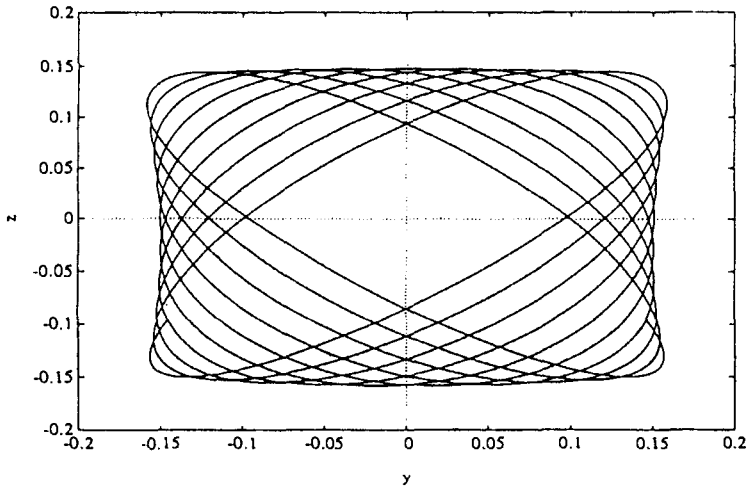


Fig. 2.6

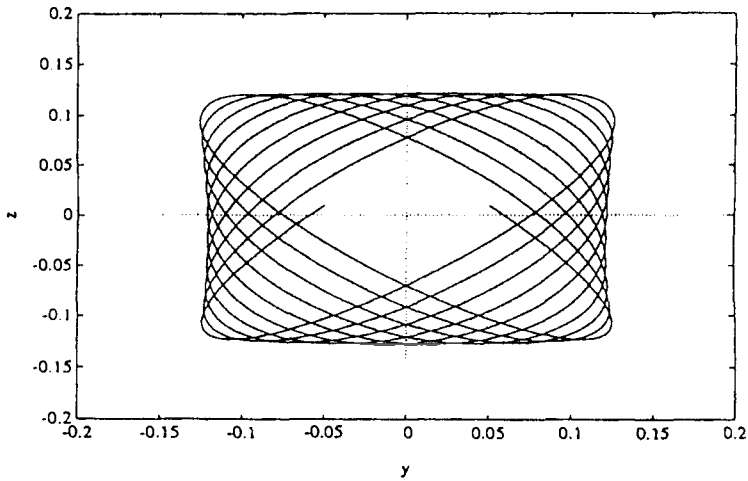


Fig. 2.7

Going backwards in time we obtain symmetrical results. Hence, in case 1 (resp.

2) we can stay in the admissible region for 29.88 units (resp. 29.74 units), equivalent to 4.75 years.

However it is not good to wait for this time to do a reasonable manoeuvre. This one should be done, in the RTBP, when $y = 0$. Using the symmetry of the solutions we can match an orbit with $(x, 0, z, \dot{x}, \dot{y}, \dot{z})$ to another with $(x, 0, z, \dot{x}, \dot{y}, -\dot{z})$ with $\Delta v = 2 |\dot{z}|$. Doing this we should do a manoeuvre with $\Delta v \simeq 143$ m/s in case 1 (121 m/s in case 2) every 4.31 years (4.30 years in case 2) to keep the spacecraft on the admissible region.

The orbits for cases 1 and 2 are shown in Figures 2.6 and 2.7.

In the real problem the quasi-periodic oscillations added to the Lissajous orbit imply that the admissible time between manoeuvres shall be a little bit less than the one in the RTBP and the Δv shall be probably slightly larger, but this has to be explored.

As a final comment, it will follow from the remaining part of this chapter that around the halo orbits there are small invariant tori. This provides also some amount of additional freedom when looking for the RTBP approximation of the nominal orbit for the SOHO mission and the related transfer orbits.

2.7 Effective Reduction to the Central Manifold

2.7.1 The Method

To start we have the Hamiltonian (2.7). First of all it should be put in the form of 2.5 by a linear canonic change and then to complex form in order to have the quadratic part as

$$\lambda q_1 p_1 + i(w_1 q_2 p_2 + w_2 q_3 p_3).$$

This is done in one step by using

$$\begin{pmatrix} x \\ y \\ y \\ p_x \\ p_y \\ p_z \end{pmatrix} = A \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix},$$

where

$$\begin{array}{ll} a_{11} &= 2\lambda\alpha, & a_{12} &= 2w_1 i\beta, \\ a_{13} &= 0, & a_{14} &= -2\lambda\alpha, \\ a_{15} &= -2w_1 i\beta, & a_{16} &= 0, \\ a_{21} &= (\lambda^2 - 1 - 2c_2)\alpha, & a_{22} &= (-w_1 - 1 - 2c_2)\beta, \end{array}$$

$$\begin{aligned}
a_{23} &= 0, & a_{24} &= (\lambda^2 - 1 - 2c_2)\alpha, \\
a_{25} &= (-w_1^2 - 1 - 2c_2)\beta, & a_{26} &= 0, \\
a_{31} &= 0, & a_{32} &= 0, \\
a_{33} &= \epsilon, & a_{34} &= 0, \\
a_{35} &= 0, & a_{36} &= 0, \\
a_{41} &= (\lambda^2 + 1 + 2c_2)\alpha, & a_{42} &= (-w_1 + 1 + 2c_2)\beta, \\
a_{43} &= 0, & a_{44} &= (\lambda^2 + 1 + 2c_2)\alpha, \\
a_{45} &= (-w_1^2 + 1 + 2c_2)\beta, & a_{46} &= 0, \\
a_{51} &= \lambda(\lambda^2 + 1 - 2c_2)\alpha, & a_{52} &= w_1 i(-w_1 + 1 - 2c_2)\beta, \\
a_{53} &= 0, & a_{54} &= -\lambda(\lambda^2 + 1 - 2c_2)\alpha, \\
a_{55} &= -w_1 i(-w_1^2 + 1 - 2c_2)\beta, & a_{56} &= 0, \\
a_{61} &= 0, & a_{62} &= 0, \\
a_{63} &= -w_2 i\epsilon, & a_{64} &= 0, \\
a_{65} &= 0, & a_{66} &= w_2 i\epsilon
\end{aligned}$$

and where λ , iw_1 , iw_2 are the eigenvalues of K at the origin, c_2 is given in (2.3) and α , β , ϵ , are:

$$\begin{aligned}
\alpha &= (2\lambda(\lambda^2(4 + 3c_2) + 4 + 5c_2 - 6c_2^2))^{-1/2}, \\
\beta &= (2w_1 i(-w_1^2(4 + 3c_2) + 4 + 5c_2 - 6c_2^2))^{-1/2}, \\
\epsilon &= (2w_2 i)^{-1/2}.
\end{aligned}$$

To generate K as a polynomial up to some degree it is better to work directly with the new complex coordinates q_j , p_j , $j = 1, 2, 3$ by using the recurrent relations. Indeed x , y , z , p_x , p_y , p_z are homogeneous linear polynomials in the variables q_j , p_j , $j = 1, 2, 3$. The quadratic part of K is easily obtained. Then to compute $Q_n(x, y, z) = \rho^n P_n(x/\rho)$, where $\rho = x^2 + y^2 + z^2$ as before, we start with

$$\begin{aligned}
Q_1(x, y, z) &= x, \\
Q_2(x, y, z) &= x^2 - \frac{1}{2}y^2 - \frac{1}{2}z^2
\end{aligned}$$

and use the recurrent relation (the one of the Legendre polynomials)

$$Q_n(x, y, z) = \left(2 - \frac{1}{n}\right) x Q_{n-1}(x, y, z) - \left(1 - \frac{1}{n}\right) (x^2 + y^2 + z^2) Q_{n-2}(x, y, z).$$

Then we want to obtain a reduction to a partial normal form (P.N.F.). We can restrict the canonical transformation to one given as the flow time one of a Hamiltonian G starting with terms of degree 3. More concretely, let M (we keep the same name) be the Hamiltonian in the (q, p) variables:

$$M = \lambda q_1 p_1 + i(w_1 q_2 p_2 + w_2 q_3 p_3) + \sum_{k \geq 3} M_k,$$

where M_k are homogeneous polynomials in (q, p) of degree k . The first part of M will be denoted by M_2 . On the other hand let

$$G = \sum_{k \geq 3} G_k(q, p),$$

and φ_t^G the time t flow using G as Hamiltonian. Define

$$(Q^T, P^T)^T = \varphi_{t=-1}^G((q^T, p^T)^T) = \varphi_{t=1}^{-G}((q^T, p^T)^T).$$

Then, as G is autonomous,

$$(q^T, p^T) = \varphi_{t=1}^G((Q^T, P^T)^T).$$

Obviously the passage from (q, p) to (Q, P) is canonical (Hori's method). Furthermore, if

$$M(q, p) = M(q(Q, P), p(Q, P)) = \widetilde{M}(Q, P),$$

one has

$$\widetilde{M}(q, p) = \varphi_{t=1}^G M(q, p) = M(q, p) + \frac{1}{1!} \frac{d}{dt} M(q, p) + \dots + \frac{1}{k!} \frac{d^k}{dt^k} M(q, p) + \dots$$

one has to compute the successive time derivatives of $M(q, p)$ along the orbits of G . This is done by using the Poisson brackets:

$$\frac{d^k}{dt^k} M(q, p) = \left[\frac{d^{k-1}}{dt^{k-1}} M(q, p), G \right] \equiv M^{(k)}(q, p).$$

First of all

$$M^{(1)}(q, p) = \sum_{k \geq 2, j \geq 3} [M_k(q, p), G_j(q, p)] = \sum_{r=3} M_r^{(1)}(q, p),$$

where

$$M_r^{(1)} = \sum_{s=2}^{r-1} [M_k, G_{r+2-s}].$$

Then

$$M^{(k)}(q, p) = \sum_{r \geq k+2} M_r^k(q, p),$$

where

$$M_r^k = \sum_{s=k+1}^{r-1} [M_s^{k-1}, G_{r+2-s}].$$

Of course $\widetilde{M}_2 = M_2$. At order 3 we have

$$\widetilde{M}_3 = M_3 + \frac{1}{1!} M_3^{(1)} = M_3 + [M_2, G_3].$$

This is the homological equation to be solved for \widetilde{M}_3 and G_3 . As M_3 is of the form

$$\sum m_{3kl} q_1^{k_1} q_2^{k_2} q_3^{k_3} p_1^{l_1} p_2^{l_2} p_3^{l_3},$$

with

$$k_i, l_i \geq 0, \quad i = 1, 2, 3, \quad \sum_{i=1}^3 (k_i + l_i) = 3,$$

we keep in \widetilde{M}_3 all the terms with $k_1 = l_1$ and try to cancel the remainder with a suitable G_3 . We note also that if a homogeneous polynomial, F , is of the form

$$F = \sum f_{k,l} q^k p^l,$$

then

$$[M_2, F] = \sum f_{k,l} (\lambda(l_1 - k_1) + iw_1(l_2 - k_2) + iw_2(l_3 - k_3)) q^k p^l.$$

Therefore we should choose

$$G_3 = \sum_{l_1 \neq k_1} m_{3kl} (\lambda(l_1 - k_1) + iw_1(l_2 - k_2) + iw_2(l_3 - k_3))^{-1} q^k p^l.$$

Here q_k denotes $q_1^{k_1} q_2^{k_2} q_3^{k_3}$ as usual, and we remark that the divisors have modulus bounded from below by $|\lambda|$.

Let us describe how to proceed to order four, the other cases being similar. We have

$$\widetilde{M}_4 = M_4 + \frac{1}{1!} M_4^{(1)} + \frac{1}{2!} M_4^{(2)}.$$

As G_3 is available we have $M_3^{(1)}$ and, hence, $M_4^{(2)} = [M_3^{(1)}, G_3]$. However $M_4^{(1)} = [M_2, G_4] + [M_3, G_3]$. Putting this together we have

$$\widetilde{M}_4 = M_4 + [M_3, G_3] + M_4^{(2)} + [M_2, G_4].$$

In the right-hand side everything is known but the last term. We write (2.7.1) as

$$\widetilde{M}_4 = F_4 + [M_2, G_4], \quad F_4 \text{ known.}$$

This is solved for \widetilde{M}_4 and G_4 as we did for (2.7.1).

In this way we obtain \widetilde{M} and G to the desired order. Having G we can compute the direct and inverse transformations from (q, p) to (Q, P) if needed. The new Hamiltonian \widetilde{M} is of the form

$$\widetilde{M} = \sum_{s \geq 2} \widetilde{M}_s, \quad \widetilde{M}_s = \sum_{|k|+|l|=s} \widetilde{m}_{skl} Q^k P^l, \quad (\text{P.N.F.})$$

with $k_1 \neq l_1$. Setting $q_1 = p_1 = 0$ we obtain the reduced Hamiltonian \widetilde{M}^0 , free from the saddle part. Finally it is convenient to go back to new real coordinates instead of the complex ones $(\overline{Q}, \overline{P})$, $\overline{Q} = (Q_2, Q_3)$, $\overline{P} = (P_2, P_3)$. They are denoted by (ξ, η) , $\xi = (\xi_1, \xi_2)$ and $\eta = (\eta_1, \eta_2)$.

We shall denote by H^0 , in accordance with Section 9, the final Hamiltonian.

2.7.2 The Program and the Results

The previous method has been implemented in the program LISWCH. The program is split in several routines which perform each one of the steps. The only input are the values of μ and the maximal order n . Several dimensions depend on n . They are

$$\begin{pmatrix} n+6 \\ 6 \end{pmatrix}, \begin{pmatrix} n+4 \\ 4 \end{pmatrix}, \begin{pmatrix} n+3 \\ 5 \end{pmatrix}, \begin{pmatrix} n+4 \\ 5 \end{pmatrix},$$

$$\begin{pmatrix} n+5 \\ 5 \end{pmatrix}, 6 \begin{pmatrix} n+6 \\ 7 \end{pmatrix} + \begin{pmatrix} n+6 \\ 6 \end{pmatrix} - 55.$$

From the input several constants are computed, like λ , w_1 , w_2 , γ and the matrix A . Also some auxiliary tables are initialized. This is done by routines CONSTA and INICI.

Then the Hamiltonian M is computed by routine GENHAM. This requires in turn the routines SUMH, ADDH, and PRODH performing elementary operations with polynomials. In these routines the homogeneous polynomials are stored in reverse lexicographical order. The routine GENLEX gives, sequentially, the reverse lexicographical order and the function LLOC says where to store terms with a given set of exponents.

The next step is to compute the P.N.F. This is done by routine FFWCH which requires the additional routine POISH1, doing the Poisson bracket of homogeneous polynomials.

When G and \widetilde{M} are available to order n one can do a check. We should have $\varphi_{t=1}^G M = \widetilde{M}$ up to order n . This is done by routine CHECK. If the changes of variables $((q, p) \text{ to } (Q, P) \text{ and vice versa})$ are required, this can be done by routine CHANGE.

Finally the routines REDHAM and AREAL perform the reduction to the central manifold and the passage to real variables.

The Table 2.5 shows the coefficients of the final reduced Hamiltonian up to order 8. There appear 5 indices before each one of the coefficients. The last 4 indices are the exponents of ξ_1 , ξ_2 , η_1 and η_2 , respectively. The first index is a check index and displays the position of the term when we order by degree and, inside every degree, by reverse lexicographical order. The final expression is quite moderate. Only 130 terms appear. To order 16 the final number of terms is 1240, but one should be aware that in the initial Hamiltonian, M , there are 38034 complex coefficients and

even in the P.N.F., \widetilde{M} , 5834 terms remain. The computing time including checks and the changes of variables takes 2^m at order 8, 1^h at order 12 and 16^h at order 16 (running on HP9000/835S). If we are only interested in the reduced Hamiltonian only 1/5 of that time is needed. We also remark that if the computations are done to order n , the changes are available at order $n - 1$.

6	2	0	0	0	.104322672764E+01	254	2	2	1	2	-.613716694201E-01
10	0	2	0	0	-.100760527574E+01	261	2	0	5	0	-.320395244659E-01
13	0	0	2	0	.104322672764E+01	263	2	0	3	2	.532436094750E-01
15	0	0	0	2	-.100760527574E+01	265	2	0	1	4	-.2099782777440E-01
18	2	0	1	0	-.651651479448E+00	269	1	5	0	1	.312121124445E-03
27	0	2	1	0	.539115445255E+00	274	1	3	2	1	-.144299000786E-01
32	0	0	3	0	.416596793211E-01	276	1	3	0	3	.295033098133E-03
36	4	0	0	0	-.857872843781E-01	283	1	1	4	1	-.902648379861E-02
40	2	2	0	0	.140437079121E+00	285	1	1	2	3	.913446913534E-02
43	2	0	2	0	.411614546444E+00	287	1	1	0	5	-.197041892894E-02
45	2	0	0	2	-.150184013209E+00	296	0	6	1	0	.339054025791E-02
50	1	1	1	1	.624904195690E-01	301	0	4	3	0	-.540500596579E-01
56	0	4	0	0	-.574685989088E-01	303	0	4	1	2	.146128108193E-01
59	0	2	2	0	-.279279638986E+00	310	0	2	5	0	.146981848249E-01
61	0	2	0	2	.124248196628E+00	312	0	2	3	2	-.278131003265E-01
66	0	0	4	0	-.265636614729E-01	314	0	2	1	4	.119054974137E-01
68	0	0	2	2	.288035154992E-01	323	0	0	7	0	.358214110312E-03
73	4	0	1	0	.696377432525E-02	325	0	0	5	2	-.667508667147E-03
77	3	1	0	1	.798359404733E-02	327	0	0	3	4	.251223700509E-04
82	2	2	1	0	-.140963190343E-01	329	0	0	1	6	.337053517668E-03
87	2	0	3	0	-.214732371131E+00	331	8	0	0	0	-.109386657286E-02
89	2	0	1	2	.144600160410E+00	335	6	2	0	0	.276360467963E-02
93	1	3	0	1	-.650849327599E-02	338	6	0	2	0	.272921610309E-01
98	1	1	2	1	-.557727541335E-01	340	6	0	0	2	.262853534821E-02
100	1	1	0	3	.960131020335E-02	345	5	1	1	1	.102317980510E-01
107	0	4	1	0	.699789175043E-02	351	4	4	0	0	-.258160126307E-02
112	0	2	3	0	.127634797175E+00	354	4	2	2	0	-.580107086110E-01
114	0	2	1	2	-.110742245281E+00	356	4	2	0	2	-.855285970237E-03
121	0	0	5	0	.917811304066E-02	361	4	0	4	0	.10570294644E+00
123	0	0	3	2	-.116332647482E-01	363	4	0	2	2	-.875032949606E-01
125	0	0	1	4	.189658326124E-02	365	4	0	0	4	.823332637879E-02
127	6	0	0	0	.524352586284E-03	370	3	3	1	1	-.151036034024E-01
131	4	2	0	0	-.489804068893E-02	377	3	1	3	1	.282370774886E-02
134	4	0	2	0	.658378938208E-01	379	3	1	1	3	-.169951976912E-02
136	4	0	0	2	-.898157155605E-02	387	2	6	0	0	.107644418777E-02
141	3	1	1	1	-.285346922409E-01	390	2	4	2	0	.420072616120E-01
147	2	4	0	0	.701867371553E-02	392	2	4	0	2	-.334432427870E-02
150	2	2	2	0	-.997998463537E-01	397	2	2	4	0	-.136791210772E+00
152	2	2	0	2	-.324336777497E-03	399	2	2	2	2	.112397054642E+00
157	2	0	4	0	.902017609399E-01	401	2	2	0	4	-.880061291889E-02
159	2	0	2	2	-.968667943688E-01	408	2	0	6	0	.830410664136E-02
161	2	0	0	4	.163202789643E-01	410	2	0	4	2	-.219231852466E-01
166	1	3	1	1	.292039913919E-01	412	2	0	2	4	.157018338398E-01
173	1	1	3	1	.298234293832E-01	414	2	0	0	6	-.150917858946E-02
175	1	1	1	3	-.137696131110E-01	419	1	5	1	1	.575975986604E-02
183	0	6	0	0	-.275110674276E-02	426	1	3	3	1	.888715186936E-02
186	0	4	2	0	.408583647863E-01	428	1	3	1	3	-.110833162962E-02
188	0	4	0	2	.824755929756E-02	437	1	1	5	1	.928800602164E-03
193	0	2	4	0	-.451046236233E-01	439	1	1	3	3	-.285933731820E-02
195	0	2	2	2	.618727561899E-01	441	1	1	1	5	.249670332593E-02
197	0	2	0	4	-.114179153085E-01	451	0	8	0	0	-.176043424562E-03
204	0	0	6	0	-.159751104192E-02	454	0	6	2	0	-.103567563622E-01
206	0	0	4	2	.356377172880E-02	456	0	6	0	2	.196274669892E-02
208	0	0	2	4	-.920455245069E-03	461	0	4	4	0	.474104611507E-01
210	0	0	0	6	-.582831778046E-03	463	0	4	2	2	-.311219684834E-01
213	6	0	1	0	-.850820746956E-03	465	0	4	0	4	.956778586635E-03
217	5	1	0	1	.446332043105E-03	472	0	2	6	0	-.304045157251E-02
222	4	2	1	0	.596989843042E-02	474	0	2	4	2	.830168071073E-02
227	4	0	3	0	-.107903835908E+00	476	0	2	2	4	-.621805667167E-02
229	4	0	1	2	.474121351924E-01	478	0	2	0	6	.868791617544E-03
233	3	3	0	1	-.747478887097E-03	487	0	0	8	0	.172298020945E-03
238	3	1	2	1	.519318080929E-02	489	0	0	6	2	-.245512765298E-03
240	3	1	0	3	-.577276345384E-03	491	0	0	4	4	.235281176498E-03
247	2	4	1	0	-.845474557118E-02	493	0	0	2	6	-.243972818512E-03
252	2	2	3	0	.147696287757E+00	495	0	0	0	8	.835853628636E-04

Table 2.5 Coefficients of the final reduced Hamiltonian up to order 8.

2.7.3 Numerical Simulations and Tests

When $H^0(\xi, \eta)$ is available we can do several simulations. Usually order 8 has been used for H^0 . Even so to compute several thousands of iterates of the Poincaré map through $\xi_2 = 0$ (equivalent to $z = 0$ in the initial variables) is a big task. This has been done for a limited set of values of the energy h ($H^0 = h$). Figures 2.8 to 2.11 (left part) show the plots of that surface of section for the energies $h = 0.1, 0.3, 0.4, 0.5, 0.7, 1.0$ and 1.5 respectively. The boundary is always the Lyapunov periodic orbit. Figure 2.11 (right part) shows the difference between the Lyapunov orbit computed on $h = 1.5$ for order 8 (continuous curve) and order 16 (dotted curve). The half widths of the figures (both in ξ_1 and η_1) are $0.36, 0.60, 0.68, 0.76, 0.92, 1.12, 1.44$ and 1.44 respectively. In all those figures we can consider the Earth sitting on the negative part of the η_1 -axis and the Sun, far away, on the positive part of the same axis. For $h < h_H \simeq 0.332820$, the Lyapunov orbit is stable. From this value on the halo orbits can be seen on the figures. It is remarkable that even for $h = 1.5$ it is hard to see stochastic zones. One can see some very small islands but the motion is quite ordered. The lines (not drawn) which separate the zone of tori (or Lissajous orbits) around the “vertical orbit” (sitting near the center of the figure) and the zones of tori around the halo orbits are the traces, on this surface of section, of the invariant manifolds of the Lyapunov orbit.

To relate the used levels of energy with the β parameter of the halo orbits one can see Table 2.6.

0.02	0.334378	0.12	0.388981
0.04	0.339052	0.14	0.409300
0.06	0.346846	0.16	0.432770
0.08	0.357762	0.18	0.459404
0.10	0.371805	0.20	0.489217

Table 2.6

Also to present some idea of the goodness of the results from a quantitative point of view we present additional results on Table 2.7. The orbits 1 to 9 have been selected by groups of 3 in each one of the levels $h = 0.5, h = 1.0, h = 1.5$. Inside each level we give the initial conditions for periodic orbits of the following types: Lyapunov (orbits 1, 4, 7), vertical (2, 5, 8) and halo (3, 6, 9). All the computations have been done at order 8. Then we transform back to Cartesian coordinates and we compare with numerical values. The maximal errors in the space variables are, in km, 16, 12, 700, 313, 238, 3 180, 1 875, 1 465 and 9 705 for orbits 1 to 9, respectively. This is improved by a factor larger than 100 for $h \leq 0.5$ and near 10 for $h \simeq 1.5$ if order 16 is used. We should pay a greater computing time and 10 Mb of memory to store the coefficients of the changes.

Orbit	Reduced	Cartesian	Numerical
1	0.0	-0.9886192288	-0.9886191198
	0.0	0.0	0.0
	0.686159	0.0	0.0
	0.0	0.0	0.0
		-0.0107659711	-0.0107660492
		0.0	0.0
2	0.0	-0.9903243964	-0.9903243149
	0.0	0.0	0.0
	0.088866	0.0	0.0
	0.698586	0.0	0.0
		-0.0007133605	-0.0007138474
		0.0100388371	0.0100387530
3	0.633035	-0.9911031366	-0.9911074139
	0.0	-0.0048042722	-0.0047996050
	-0.015524	0.0	0.0
	0.292145	-0.0029410293	-0.0029333451
		0.0017601018	0.0017917750
		0.0042457275	0.0042520908
4	0.0	-0.9881865512	-0.988184666
	0.0	0.0	0.0
	0.968531	0.0	0.0
	0.0	0.0	0.0
		-0.0150668946	-0.0150683874
		0.0	0.0
5	0.0	-0.9906603953	-0.9906588061
	0.0	0.0	0.0
	0.179863	0.0	0.0
	0.979163	0.0	0.0
		-0.0015091839	-0.0015181117
		0.01423098598	0.0142287520
6	0.806790	-0.9919717435	-0.9919730051
	0.0	-0.0059292421	-0.0059080613
	0.017889	0.0	0.0
	0.572802	-0.0037342856	-0.0037259701
		0.0026423380	0.0026711102
		0.0085015587	0.0085389294

Orbit	Reduced	Cartesian	Numerical
7	0.0	-0.9878719651	-0.9878594540
	0.0	0.0	0.0
	1.184712	0.0	0.0
	0.0	0.0	0.0
		-0.0183518934	-0.0183610059
		0.0	0.0
8	0.0	-0.99909942212	-0.9909844530
	0.0	0.0	0.0
	0.271440	0.0	0.0
	1.188193	0.0	0.0
		-0.0023749889	-0.0024266444
		0.0174669454	0.0174506786
9	0.963854	-0.9929400824	-0.9929183465
	0.0	-0.0068042692	-0.0067395456
	0.050572	0.0	0.0
	0.739605	-0.0043961422	-0.0044301972
		0.0037697329	0.0037048142
		0.0112519581	0.0113758971

Table 2.7

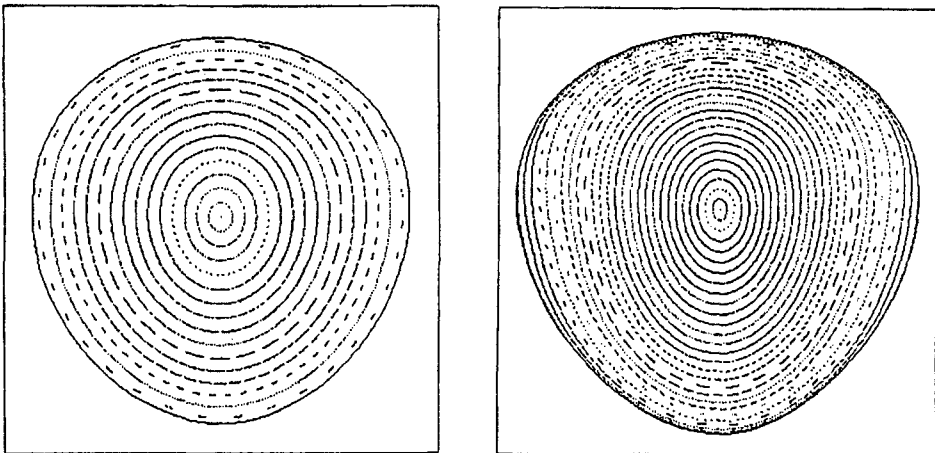


Fig. 2.8

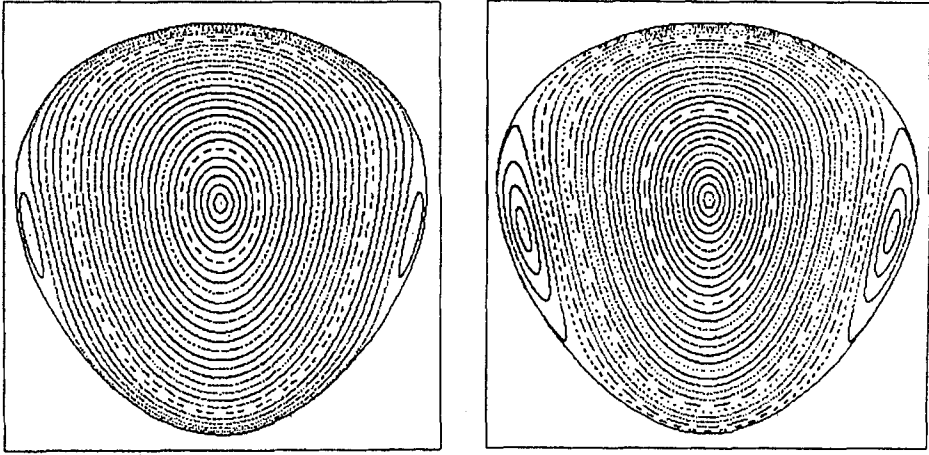


Fig. 2.9

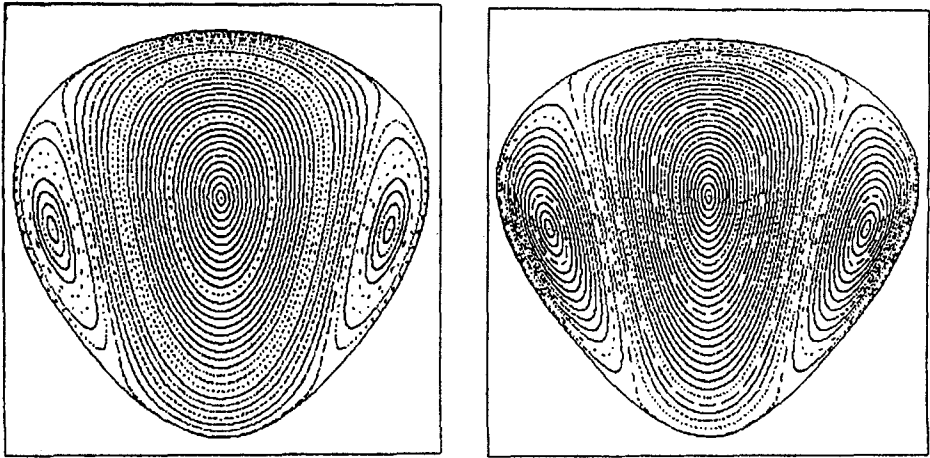


Fig. 2.10

2.8 Conclusions

So far we have seen that it is possible to describe quite precisely by analytic means the neighborhood of L_1 for the Earth-Moon problem as a RTBP skipping the unstable/stable directions up to distances less than 0.9×10^6 km from the Earth. Indeed the Lyapunov orbit for $h = 1.5$ ranges from less than 0.9×10^6 to more than 1.85×10^6 km from the Earth and its range in the y variable attains 2.9×10^6 km. The use of the central manifold to high order is much better than the methods used

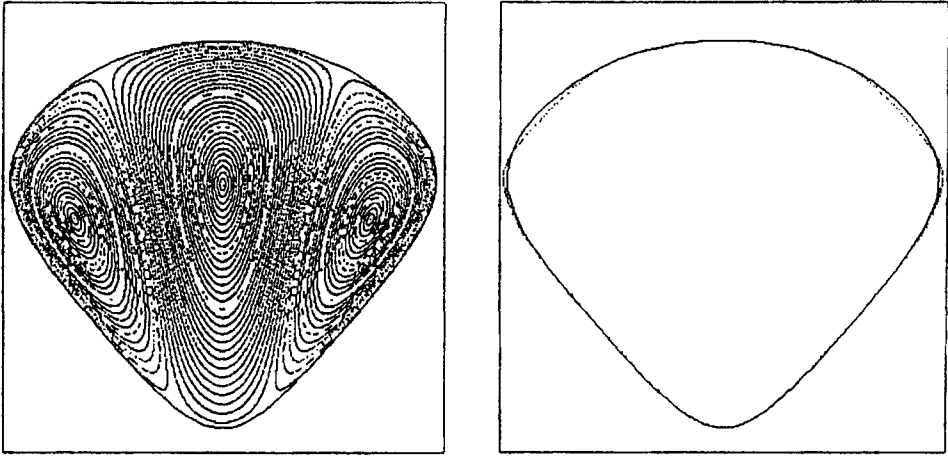


Fig. 2.11

before to obtain Lissajous orbits. It is not affected by small divisors problems. It has the problem that it does not produce explicit solutions. The description of the neighborhood could not be done without skipping the instability.

To go back to the real problem one should refer again to parallel shooting including the perturbations starting with a numerical solution of M^0 . Eventually continuation should be used for large amplitudes.

This page is intentionally left blank

Chapter 3

Quasi-periodic Halo Orbits

3.1 Numerical Refinement

Taking as an starting point the analytic quasi-periodic halo orbit obtained by program QPO, in the way described in Chapter VII of [2], a quasi-periodic orbit of the real solar system is computed, using the JPL ephemeris, by means of a parallel shooting procedure.

In the normalized reference system, see Appendix B or Chapter V of [2], we say that a qpo has z -amplitude β if it comes from improving a halo orbit of z -amplitude β .

If $(t, x, y, z, \dot{x}, \dot{y}, \dot{z})$ denote the time, position and velocity in the normalized system, we define the Poincaré map, \mathcal{P} , with the surface of section $y = 0$, following the flow when t increases:

$$\begin{aligned} \mathcal{P}(t, x, y = 0, z, \dot{x}, \dot{y}, \dot{z}) &= (\bar{t}, \bar{x}, \bar{z}, \dot{\bar{x}}, \dot{\bar{y}}, \dot{\bar{z}}) \\ &= (\mathcal{P}_1(t, x, z, \dot{x}, \dot{y}, \dot{z}), \dots, \mathcal{P}_6(t, x, z, \dot{x}, \dot{y}, \dot{z})). \end{aligned}$$

The surface of section $y = 0$ gives the intervals at which the parallel shooting is defined, in the sense that each interval of the procedure is comprised between two crosses with this surface of section.

We denote by $Q_i = (t_i, x_i, z_i, \dot{x}_i, \dot{y}_i, \dot{z}_i)$ the i th point in the partition of the full interval at which the qpo shall be computed. If the parallel shooting is split in N subintervals then $i = 0, \dots, N$.

So, in principle, we have $6(N + 1)$ free variables. One should take into account that we choose the initial epoch at which the qpo must be computed, and, in addition, we improve a qpo with a certain z -amplitude. So we fix t_0 and z_0 at the beginning of the procedure and the number of free variables (the components of Q) is reduced to $6N + 4$.

In this way we need $6N + 4$ equations for the determination of these variables.

Any parallel shooting must satisfy the matching conditions:

$$F_{6i+j}(Q) = \mathcal{P}_j(Q_i) - (Q_{i+1})_j = 0, \quad j = 1, \dots, 6, \quad i = 0, \dots, N-1,$$

where F is the vector of $6N + 4$ equations to be fulfilled by the variable Q .

The matching conditions give us $6N$ equations. The remaining four ones can be imposed in several ways obtaining different approaches of the qpo.

The ones that we have taken are:

$$\begin{aligned} F_{6N+1} &= x_0 - x_0^t - (x_N - x_N^t) &= 0, \\ F_{6N+2} &= z_N - z_N^t &= 0, \\ F_{6N+3} &= \pi_4(\dot{x}_0 - \dot{x}_0^t) + \pi_5(\dot{y}_0 - \dot{y}_0^t) + \pi_6(\dot{z}_0 - \dot{z}_0^t) &= 0, \\ F_{6N+4} &= \pi_4(\dot{x}_N - \dot{x}_N^t) + \pi_5(\dot{y}_N - \dot{y}_N^t) + \pi_6(\dot{z}_N - \dot{z}_N^t) &= 0, \end{aligned}$$

Here the superindex t denotes a value taken from the analytic quasi-periodic orbit and π_4, π_5, π_6 are the three last components of the projection factors in the unstable component of the quasi-periodic orbit (see [2]).

As a summary, the system of equations to be solved is:

$$\begin{aligned} F_1(Q) &= \mathcal{P}_1(t_0, x_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0) - t_1 &= 0, \\ F_2(Q) &= \mathcal{P}_2(t_0, x_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0) - x_1 &= 0, \\ &\dots &\dots\dots \\ F_6(Q) &= \mathcal{P}_6(t_0, x_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0) - \dot{z}_1 &= 0, \\ F_7(Q) &= \mathcal{P}_1(t_1, x_1, z_1, \dot{x}_1, \dot{y}_1, \dot{z}_1) - t_2 &= 0, \\ &\dots &\dots\dots \\ F_{6N}(Q) &= \mathcal{P}_6(t_{N-1}, x_{N-1}, z_{N-1}, \dot{x}_{N-1}, \dot{y}_{N-1}, \dot{z}_{N-1}) - \dot{z}_N &= 0, \\ F_{6N+1}(Q) &= x_0 - x_0^t - (x_N - x_N^t) &= 0, \\ F_{6N+2}(Q) &= z_N - z_N^t &= 0, \\ F_{6N+3}(Q) &= \pi_4(\dot{x}_0 - \dot{x}_0^t) + \pi_5(\dot{y}_0 - \dot{y}_0^t) + \pi_6(\dot{z}_0 - \dot{z}_0^t) &= 0, \\ F_{6N+4}(Q) &= \pi_4(\dot{x}_N - \dot{x}_N^t) + \pi_5(\dot{y}_N - \dot{y}_N^t) + \pi_6(\dot{z}_N - \dot{z}_N^t) &= 0, \end{aligned}$$

where

$$Q = (t_0, x_0, \dot{x}_0, \dot{y}_0, \dot{z}_0, t_1, x_1, z_1, \dot{x}_1, \dot{y}_1, \dot{z}_1, \dots, t_N, x_N, z_N, \dot{x}_N, \dot{y}_N, \dot{z}_N).$$

Denoting the equations by $F(Q) = 0$ and taking $Q^{(0)}$ as the initial values of the variables, the improved values are obtained by means of a Newton procedure:

$$DF(Q^{(j)})(Q^{(j+1)} - Q^{(j)}) = -F(Q^{(j)}).$$

3.1.1 Computation of the Initial Values

The initial values of the components of $Q^{(0)}$, t_0 and z_0 can be obtained as follows. For a halo orbit it is clear that, belonging to a one-parameter family, it is enough to fix a value of z when $y = 0$ at a given epoch. From the characteristic curve of the family, analytically expressed by $\Delta(\alpha, \beta) = 0$, the x -amplitude, α , can be

obtained from the z -amplitude, β . It is true that β does not coincide with z_0 , but the expression of z when the halo angle is zero: $z = \beta + O_2(\alpha, \beta)$, where O_2 means terms of second order in α, β , shows that z and β are close. An iterative procedure can be used to determine β .

The terms added to the halo part of the orbit by program QP0 (see Chapter VII of [2]) depend on α, β and t .

Hence, for a given t_0 , the values of $x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0$ depend on α and β , and α and β should satisfy the Δ condition. They also depend on the epoch, t^* , at which the halo part of the qpo crosses the $y = 0$ hyperplane with z having the sign assigned to β . If the quasi-periodic perturbations were zero, the time t^* will coincide with t_0 (modulus the period of the halo orbit) if the sign of β and z_0 agree. Otherwise t_0 and t^* differ in a half period (also modulus the period).

When the quasi-periodic part of the orbit is included t_0 and t^* are close but they are not forcedly equal (or they differ roughly in a half period of the halo part).

First of all we have used an iterative procedure to determine β and t^* from z_0 and t_0 . This is implemented in routine ESTQP0. We proceed as follows:

- We start with an initial value of the parameters β and t^* . If nothing better is available we use $\beta = \pm z_0$, $t^* = t_0 + \delta \cdot T/2$ where the sign of β is chosen in order to agree with the previously assigned sign. In the expression of t^* the parameter δ is set equal to zero if β and z_0 have the same sign, and $\delta = 1$ otherwise. T refers to the approximate period of the halo orbit. As α and β are unknown, this period is not available, however taking into account that we deal with small and medium size halo orbits and its period has not large changes along the family, the approximate value of T is the one which corresponds to $\beta = 0.08$.
- When an approximation of (β, t^*) is available, we compute (using routine ANAQP0) the corresponding values of y and z (and in fact all the coordinates). The standing equations are $y(\beta, t^*, t_0) = 0$ and $z(\beta, t^*, t_0) = z_0$.
- The previous equations are solved by Newton's method. For this purpose we need the partial derivatives: $\frac{\partial y}{\partial \beta}$, $\frac{\partial y}{\partial t^*}$, $\frac{\partial z}{\partial \beta}$ and $\frac{\partial z}{\partial t^*}$. They are computed using numerical differentiation with an step ε , selected equal to 10^{-6} . Using these partial derivatives the Newton procedure is iterated till convergence. The value 10^{-11} in the corrections of β and t^* have been used as stopping criteria.

The full initial conditions, $(x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0)$ at $t = t_0$, are known once the values of β and t^* have been determined. Now we need to determine, for the analytic qpo, the epochs t_1, \dots, t_N of the successive passages through $y = 0$. The separation $t_{i+1} - t_i$ is a "half revolution", so when t_i is known, we determine an approximate value of t_{i+1} adding $T/2$ to t_i . Then the routine PREDY0 determines the corrections to be applied to t_{i+1} in order to get $y = 0$. This is done using again Newton's method. In this case the required derivative, \dot{y} , is available from routine ANAQP0.

By iteration (up to an error less than 10^{-10}) of $\Delta t_{i+1} = -y(t_{i+1})/\dot{y}(t_{i+1})$ we obtain the value t_{i+1} .

3.1.2 The Differential Matrix $DF(Q)$

The differential matrix $DF(Q)$ has the following structure:

$$DF(Q) = \begin{pmatrix} A_0 & -I & & & & \\ & A_1 & -I & & & \\ & & A_2 & -I & & \\ & & & \dots & \dots & \\ & & & & \dots & \dots \\ & & & & & A_{N-1} & -I \\ B_N & & & & & & A_N \end{pmatrix},$$

where A_i , $i = 1, 2, \dots, N$ denote the 6×6 matrices $DP(Q_i)$, and A_0 is the 6×4 matrix obtained from $DP(Q_0)$ skipping the columns 1 and 3 (related to t_0 and z_0 that do not change).

The matrix B_N is 4×4 , with all entries equal zero except the element in the first row and column which is set equal to 1.

The matrix A_N is a 4×6 matrix with the following structure:

$$A_N = \begin{pmatrix} a_1 & -1 & a_5 & 0 & 0 & 0 \\ a_2 & 0 & a_6 & -1 & 0 & 0 \\ a_3 & 0 & a_7 & 0 & -1 & 0 \\ a_4 & 0 & a_8 & 0 & 0 & -1 \end{pmatrix},$$

where $a_1 = \frac{\partial x_N^t}{\partial t_N}$, $a_2 = \frac{\partial \dot{x}_N^t}{\partial t_N}$, $a_3 = \frac{\partial y_N^t}{\partial t_N}$, $a_4 = \frac{\partial \dot{z}_N^t}{\partial t_N}$, $a_5 = \frac{\partial x_N^t}{\partial z_N}$, $a_6 = \frac{\partial \dot{x}_N^t}{\partial z_N}$, $a_7 = \frac{\partial y_N^t}{\partial z_N}$ and $a_8 = \frac{\partial \dot{z}_N^t}{\partial z_N}$.

The derivatives a_1, \dots, a_8 have been computed using numerical differentiation.

3.1.3 Computing the Poincaré Map and its Differential

Let $\vec{r}, \vec{r}_1, \dots, \vec{r}_k$ the ecliptic coordinates (centered at the center of masses of the solar system) of the spacecraft and of the bodies number 1 to k of the solar system. The lack of coherence and radiation pressure are included in the masses of the corresponding bodies.

Then the equations of motion are:

$$\ddot{\vec{r}} = - \sum_{i=1}^k Gm_i \frac{\vec{r} - \vec{r}_i}{d_i^3},$$

where m_i is the (total) mass of the i th body, G is the gravitational constant and $d_i = |\vec{r} - \vec{r}_i|_2$.

The Poincaré map is expressed in normalized coordinates. Hence, given initial conditions $t, x, y = 0, z, \dot{x}, \dot{y}, \dot{z}$, they are converted to days and ecliptic position and velocity. Then the numerical integration of Newton's equations is started. First we perform the integration for a time interval which is near to, but a little bit less than, the half period or the full period of the halo part of the analytic qpo, depending of the type of intervals that we have chosen. Then, after each step in the numerical integration, the point is transformed back to normalized time and coordinates and the condition $y = 0$ is searched. When, at some step, a change of sign of y is detected, a Newton's method is used through $\Delta t = -y/\dot{y}$ to obtain the final time, $\mathcal{P}_1(t, x, z, \dot{x}, \dot{y}, \dot{z})$, under the Poincaré map. The remaining variables, $x, z, \dot{x}, \dot{y}, \dot{z}$ give the other five components of the image of the Poincaré map.

To obtain the differential, $D\mathcal{P}$, of the Poincaré map, the equations of motion should be integrated simultaneously with the variational equations. As the system is nonautonomous we have also variations with respect to the initial time. First we look in ecliptic coordinates. Let $x_1, x_2, x_3, x_4, x_5, x_6$ denote ecliptic position and velocity.

The equations of the autonomized system are:

$$\begin{aligned} \dot{x}_0 &= 1, \\ \dot{x}_1 &= x_4, \quad \dot{x}_2 = x_5, \quad \dot{x}_3 = x_6, \\ \dot{x}_4 &= -\sum_{i=1}^k Gm_i \frac{x_1 - x_{i,1}}{d_i^3}, \\ \dot{x}_5 &= -\sum_{i=1}^k Gm_i \frac{x_2 - x_{i,2}}{d_i^3}, \\ \dot{x}_6 &= -\sum_{i=1}^k Gm_i \frac{x_3 - x_{i,3}}{d_i^3}. \end{aligned}$$

The variational equations can be written in the form $\dot{V} = D \cdot V$, where V and D are 7×7 matrices and V is initialized to the identity matrix. The matrix D has the block structure:

$$D = \left(\begin{array}{c|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & 1 & 0 & 0 \\ 0 & & \bigcirc & & 0 & 1 & 0 \\ 0 & & & & 0 & 0 & 1 \\ \hline g_1 & h_{11} & h_{12} & h_{13} & 0 & 0 & 0 \\ g_2 & h_{21} & h_{22} & h_{23} & 0 & 0 & 0 \\ g_3 & h_{31} & h_{32} & h_{33} & 0 & 0 & 0 \end{array} \right),$$

where:

$$\begin{aligned}
g_1 &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(-x_{i,1} + \frac{3q_i (x_1 - x_{i,1})}{d_i^2} \right), \\
g_2 &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(-x_{i,2} + \frac{3q_i (x_2 - x_{i,2})}{d_i^2} \right), \\
g_3 &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(-x_{i,3} + \frac{3q_i (x_3 - x_{i,3})}{d_i^2} \right), \\
q_i &= (x_1 - x_{i,1})\dot{x}_{i,1} + (x_1 - x_{i,2})\dot{x}_{i,2} + (x_1 - x_{i,3})\dot{x}_{i,3}, \\
h_{11} &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(1 - \frac{3(x_1 - x_{i,1})^2}{d_i^2} \right), \\
h_{12} &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(-\frac{3(x_1 - x_{i,1})(x_2 - x_{i,2})}{d_i^2} \right), \\
h_{13} &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(-\frac{3(x_1 - x_{i,1})(x_3 - x_{i,3})}{d_i^2} \right), \\
h_{21} &= h_{12}, \\
h_{22} &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(1 - \frac{3(x_2 - x_{i,2})^2}{d_i^2} \right), \\
h_{23} &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(-\frac{3(x_2 - x_{i,2})(x_3 - x_{i,3})}{d_i^2} \right), \\
h_{31} &= h_{13}, \\
h_{32} &= h_{23}, \\
h_{33} &= -\sum_{i=1}^k \frac{Gm_i}{d_i^3} \left(1 - \frac{3(x_3 - x_{i,3})^2}{d_i^2} \right).
\end{aligned}$$

In principle the variational equations would be 49 but, due to the fact that the first row of D is zero, the full first row of the variational variables must remain constant during the integration. We can skip seven equations plus the one: $\dot{t} = \dot{x}_0 = 1$. So the variational equations can be written as:

$$\begin{pmatrix} x_7 \\ x_8 \\ \dots \\ \dots \\ x_{12} \end{pmatrix}^\bullet = \tilde{D} \begin{pmatrix} x_7 \\ x_8 \\ \dots \\ \dots \\ x_{12} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ g_1 \\ g_2 \\ g_3 \end{pmatrix},$$

$$\begin{pmatrix} x_{13} & x_{19} & \cdots & \cdots & x_{43} \\ x_{14} & x_{20} & \cdots & \cdots & x_{44} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{18} & x_{24} & \cdots & \cdots & x_{48} \end{pmatrix}^{\bullet} = \tilde{D} \begin{pmatrix} x_{13} & x_{19} & \cdots & \cdots & x_{43} \\ x_{14} & x_{20} & \cdots & \cdots & x_{44} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{18} & x_{24} & \cdots & \cdots & x_{48} \end{pmatrix},$$

where:

$$\tilde{D} = \left(\begin{array}{ccc|ccc} & & & 1 & 0 & 0 \\ & \bigcirc & & 0 & 1 & 0 \\ & & & 0 & 0 & 1 \\ h_{11} & h_{12} & h_{13} & 0 & 0 & 0 \\ h_{21} & h_{22} & h_{23} & 0 & 0 & 0 \\ h_{31} & h_{32} & h_{33} & 0 & 0 & 0 \end{array} \right).$$

The initial conditions are $x_j = 0$ for $j = 1$ to 48 except $x_{13} = x_{20} = \dots = x_{48} = 1$. When the integration of the variational matrix, V , is finished one should do two things:

- (1) To obtain the differential matrix of the Poincaré map, \mathcal{P} , in normalized coordinates.
- (2) To obtain, also in normalized coordinates, the matrix of the partial derivatives of $x, y, z, \dot{x}, \dot{y}$ and \dot{z} at the time t_{i+1} with respect to the values of $x, y, z, \dot{x}, \dot{y}, \dot{z}$ at $t = t_i$. In the case of a halo orbit this matrix would be the monodromy matrix.

The second point is not used by the parallel shooting algorithm, but will be used in the course of the determination of the numerical projection factors.

First of all we require to express the 7×7 matrix V in normalized coordinates, V_n . We have $V_n = DT_2 \cdot V \cdot DT_1$, where DT_1 is the differential of the transformation, T_1 , from normalized variables to ecliptic ones,

$$T_1(t, x, y, z, \dot{x}, \dot{y}, \dot{z}) = (t_d, x_1, x_2, x_3, x_4, x_5, x_6),$$

evaluated at the initial point, and DT_2 is the differential of the inverse transformation, T_2 , at the final time,

$$T_2(\bar{t}_d, \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6) = (\bar{t}, \bar{x}, \bar{y}, \bar{z}, \dot{\bar{x}}, \dot{\bar{y}}, \dot{\bar{z}}).$$

The overlined variables mean the image variables, and t_d refers to the time expressed in Julian days instead of adimensional (normalized) time units.

Using the routines **TRANSLI**, to go from normalized to adimensional and **TRANS**, to go from adimensional to ecliptic, the transformation T_1 is obtained. T_2 is given by the inverse.

The differentials DT_1 , DT_2 have been evaluated by numerical differentiation. The step used in the variables is a magnitude given by the user. We have used the value 10^{-6} as more convenient.

When V_n is available, the matrix referred to in 2) is obtained easily deleting the first row and column. The matrix of 1) requires a little more work. We put

$$V_n \begin{pmatrix} \Delta t \\ \Delta x \\ 0 \\ \Delta z \\ \Delta \dot{x} \\ \Delta \dot{y} \\ \Delta \dot{z} \end{pmatrix} + \delta t \begin{pmatrix} 1 \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix}_f = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix}.$$

The amount δt is left free in order to have $c_2 = 0$. The components of the vector field at the last point, in normalized coordinates, $(\dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})_f$, are obtained again from routines **TRANS** and **TRANSLI**.

If the element (i, j) of the matrix V_n is denoted by V_{nij} , $i, j = 0, \dots, 6$, δt is determined through:

$$V_{n20}\Delta t + V_{n21}\Delta x + V_{n23}\Delta z + V_{n24}\Delta \dot{x} + V_{n25}\Delta \dot{y} + V_{n26}\Delta \dot{z} + \delta t \cdot \dot{y} = 0.$$

If this value of δt is substituted in the expressions of c_0, c_1, \dots, c_6 , we get:

$$\begin{aligned} c_0 &= V_{n00}\Delta t + V_{n01}\Delta x + V_{n03}\Delta z + V_{n04}\Delta \dot{x} + V_{n05}\Delta \dot{y} + V_{n06}\Delta \dot{z} \\ &\quad - \left(\frac{1}{\dot{y}} \right)_f (V_{n20}\Delta t + V_{n21}\Delta x + V_{n23}\Delta z + V_{n24}\Delta \dot{x} + V_{n25}\Delta \dot{y} \\ &\quad + V_{n26}\Delta \dot{z}) \\ &= \left(V_{n00} - \frac{V_{n20}}{\dot{y}} \right)_f \Delta t + \dots + \left(V_{n06} - \frac{V_{n26}}{\dot{y}} \right)_f \Delta \dot{z}, \\ &\vdots \\ c_6 &= V_{n60}\Delta t + V_{n61}\Delta x + V_{n63}\Delta z + V_{n64}\Delta \dot{x} + V_{n65}\Delta \dot{y} + V_{n66}\Delta \dot{z} \\ &\quad - \left(\frac{\ddot{z}}{\dot{y}} \right)_f (V_{n20}\Delta t + V_{n21}\Delta x + V_{n23}\Delta z + V_{n24}\Delta \dot{x} + V_{n25}\Delta \dot{y} \\ &\quad + V_{n26}\Delta \dot{z}) \\ &= \left(V_{n60} - \frac{\ddot{z}V_{n20}}{\dot{y}} \right)_f \Delta t + \dots + \left(V_{n66} - \frac{\ddot{z}V_{n26}}{\dot{y}} \right)_f \Delta \dot{z}. \end{aligned}$$

The components of c_0, c_1, \dots, c_6 (seen as linear forms) give the matrix DP :

$$\begin{aligned}
 (DP)_{11} &= \left(V_{n00} - \frac{V_{n20}}{\dot{y}} \right)_f, \\
 &\vdots \\
 (DP)_{66} &= \left(V_{n66} - \frac{\ddot{z}V_{n26}}{\dot{y}} \right)_f.
 \end{aligned}$$

The computation of the image under the Poincaré map and its differential is done in the routine **P0INC2**. The routine **DERIVP** contains the vector field in dimension 48 obtained by adding to the equations of motion the variational equations (we remark that the equation $\dot{x}_0 = 1$ and the related variational equations, $\dot{V}_{0i} = 0$, $i = 0, \dots, 6$ can be skipped).

With the initial assignments of values to Q and the iterative procedure, the parallel shooting program **GQPTPS** finds a final value of Q .

3.2 Main Program and Basic Routines

The program which performs the computations of the quasi-periodic halo orbits makes use of a large set of auxiliary routines, numerical integrator, manipulation of formal series, changes of coordinates, routines for the use of the JPL ephemeris, etc. which are not explained here (see Appendix A for its description). Only the ones that are closely related to the problem are described below.

Program GQPTPS This program produces the initial conditions, at a selected number of points, of the motion of the spacecraft in the solar system near a quasi-periodic halo orbit. The starting data are obtained from the output of programs **ANACOM**, **QPO** and **CONA**. (See [2] for these last three programs).

Subroutine ANAQPO Computes the coordinates (position, velocity and acceleration) of a quasi-periodic halo orbit according to the analytic theory using the output of program **QPO**.

Subroutine DERIVP This routine computes the ecliptic vector field defining the equations of motion and the variational equations, with respect to time and to coordinates in the phase space.

Subroutine ESTQPO When the quasi-periodic part of the halo orbit is included, this routine performs the iterative procedure to compute β and t^* from z_0 and t_0 .

Subroutine MAGCON This routine computes analytic approximations, by using the RTBP model, of the normalized projection factors on the unstable direction which appear in the last three equations that define the parallel shooting procedure. For its computation it uses the output of program **CONA**.

Subroutine POINC2 This routine computes explicitly the Poincaré map of the flow with the surface of section $y = 0$. The differential of the Poincaré map can also be computed making use of numerical differentiation.

Subroutine PREDY0 When the values of β and t^* have been obtained, this routine computes the epochs t_1, \dots, t_N of the successive passages of the quasi-periodic halo orbit through $y = 0$.

Subroutine TPSS2 This is one of the main routines of the program. Making use of the Poincaré map (routine POINC2) it performs the computations of the parallel shooting algorithm.

3.2.1 Numerical Results

An extensive use of this program has been done in the study of the transfer problem. Here we present, as a sample of the typical output of the program, the results corresponding to a halo orbit of z -amplitude $\beta=0.08$.

The numerical quasi-periodic orbit has been refined starting from a periodic halo orbit with these characteristics:

- The spacecraft section divided by the mass (SPCSM = 0.01).
- The full solar system is considered (IPA).
- The orbit is computed in the Earth+Moon-Sun System (IC0=2).
- The z -amplitude (Z0) is 0.08.
- The initial epoch, in modified Julian days starting at 1950.0, (T0) at which the integration is started is 16587.0 (1-VI-1995).
- The number of half revolutions (NIT) for the parallel shooting is 40.
- The parallel shooting was performed using half revolutions instead of full revolutions (NHR=1). It is possible to do the computations of the parallel shooting using full revolutions (NHR=2) but some care must be taken when using the output of this program to do the transfer computations.
- The parameter β has been chosen with the same sign as z_0 (IBETA=1), it can be chosen with the opposite sign taking (IBETA=-1).
- The bound for the error in the equations for a stopping criteria in the parallel shooting procedure (BOUND1) is 2×10^{-9} .
- The bound for the error in the corrections for a stopping criteria in the parallel shooting procedure (BOUND2) is 3×10^{-10} .
- The selected number of maximum iterations of the procedure (NUMITM) is 12.
- The step for the numerical differentiation for computing the Poincaré map (EPS) is 10^{-6} .

With these data program GQPTPS has needed 6 iterations having as final error in the equations of the parallel shooting the value $2.8927 \cdot 10^{-10}$.

Only a sample of the output is given here. It has been split in two parts. In the

first one we give the normalized coordinates of the intersections of the numerical quasi-periodic orbit with the surface of section $y = 0$ (VECTOR XS). In the second part we give the epochs at which these intersections take place in different units: normalized, modified Julian dates and calendar dates.

The output of the program gives also the monodromy matrix related to each interval of the parallel shooting. This part of the output is not included here because of its large extension.

Finally some plots of the projections of the qpo on the coordinate planes are shown in Figure 3.1.

```

SPCSM = .10000000000E-01   IPA = 1 1 11 1 1 1 1 1 1 1 1 0 0 0 ICD = 2
ZO = .080000 TO= 16587.00000 NIT = 40 NHR = 1 IBETA = 1
BOUND1 = .2000E-08 BOUND2 = -.3000E-09 NUMITM = 12 EPS = .1000E-05

VECTOR XS (normalized coordinates)
0 -.11600476054E+00 .00E+00 .80000000002E-01 .10826885389E-01 .86029830892E+00 -.69048017066E-02
1 .15892844037E+00 .71E-11 -.651133755450E-01 .22149457655E-02 -.94301403284E+00 .69486923151E-02
2 -.11488379204E+00 -.47E-11 .81450254119E-01 -.89261259915E-03 .90153843583E+00 -.52032939411E-02
3 .16133690121E+00 .71E-11 -.65702194104E-01 -.10612061403E-02 -.98132463978E+00 .65523734005E-02
4 -.11210263903E+00 -.47E-11 .81319829491E-01 .10077074844E-02 .85837949759E+00 -.51095675728E-02
5 .15875780152E+00 .71E-11 -.66154478048E-01 .17944442095E-02 -.93902846371E+00 .57145011265E-02
6 -.11456965483E+00 -.47E-11 .82684525140E-01 -.10718417334E-02 .89780324139E+00 -.40709123115E-02
7 .16152773706E+00 .71E-11 -.66687324460E-01 -.10919297659E-02 -.98476483863E+00 .51099377425E-02
8 -.11221793799E+00 -.47E-11 .32465369173E-01 .10921878703E-02 .86124584359E+00 -.39985627301E-02
9 .15859763506E+00 .71E-11 -.66956720297E-01 .13206980203E-02 -.93574644182E+00 .42243943161E-02
10 -.11426249898E+00 -.47E-11 .83604828979E-01 -.11888138236E-02 .89369090144E+00 -.27700318156E-02
11 .16167939178E+00 .71E-11 -.67403115761E-01 -.96095979396E-03 -.98751665718E+00 .35238311800E-02
12 -.11240079010E+00 -.47E-11 .83282018787E-01 .11730051798E-02 .86490253561E+00 -.27747480684E-02
13 .15845038527E+00 .71E-11 -.67490295341E-01 .65950783055E-03 -.93286474582E+00 .26160143185E-02
14 -.11394377662E+00 -.47E-11 .84166673428E-01 -.13230253833E-02 .88915157632E+00 -.14127633162E-02
15 .16176834292E+00 .71E-11 -.67833447511E-01 -.54665877571E-03 -.98956041436E+00 .18924410125E-02
16 -.11264708626E+00 -.47E-11 .83750985460E-01 .12530303237E-02 .86910863613E+00 -.15019312234E-02
17 .15834003364E+00 .71E-11 -.67749090235E-01 .14091935384E-03 -.93074579702E+00 .98966776381E-03
18 -.11363714076E+00 -.47E-11 .84342373598E-01 -.14447564886E-02 .88471692457E+00 -.60403509874E-04
19 .16189117103E+00 .71E-11 -.67953551019E-01 -.23254589803E-03 -.99130021513E+00 .18763826273E-03
20 -.11294874492E+00 -.48E-11 .03948304224E-01 .13334295927E-02 .87373859269E+00 -.17858473250E-03
21 .15827977856E+00 .71E-11 -.67692637286E-01 -.20529937839E-03 -.92968695079E+00 -.60153781054E-03
22 -.11336254383E+00 -.47E-11 .84159845749E-01 -.14679523051E-02 .88030368191E+00 .12125124118E-02
23 .16199059998E+00 .71E-11 -.67809710392E-01 -.17326171870E-03 -.99221367929E+00 -.15422177274E-02
24 -.11325912147E+00 -.47E-11 .83628267463E-01 .13800825620E-02 .87850081443E+00 .11556109430E-02
25 .15829291749E+00 .71E-11 -.67392092348E-01 -.34437214383E-03 -.92969957462E+00 -.21126756444E-02
26 -.11312447640E+00 -.47E-11 .83657039284E-01 -.13946197659E-02 .87600979951E+00 .24228106890E-02
27 .16199461490E+00 .71E-11 -.67419585262E-01 -.95484639992E-04 -.99191514333E+00 -.31876596287E-02
28 -.11363627707E+00 -.47E-11 .83105948375E-01 .13832765905E-02 .88340121432E+00 .24447722180E-02
29 .15034192342E+00 .71E-11 -.66846833223E-01 -.38726443234E-03 -.93069117942E+00 -.35359927912E-02
30 -.11289069014E+00 -.47E-11 .82859357966E-01 -.12253868661E-02 .87172823008E+00 .35354637929E-02
31 .16187291785E+00 .71E-11 -.66798036112E-01 .31896237576E-03 -.99042237417E+00 -.46302090325E-02
32 -.11397232338E+00 -.47E-11 .82325376017E-01 .13143150340E-02 .88813676070E+00 .36581376418E-02
33 .15841542208E+00 .71E-11 -.66087227991E-01 -.54862858291E-03 -.93223519971E+00 -.40791893586E-02
34 -.11268585717E+00 -.47E-11 .81811500073E-01 -.10822256651E-02 .86753930897E+00 .45574443249E-02
35 .16174059152E+00 .71E-11 -.65941607626E-01 .10471959547E-02 -.98856658345E-03 -.59479554518E-02
36 -.11433852262E+00 -.48E-11 .81277120631E-01 .12368362067E-02 .89299670732E+00 .48000109949E-02
37 .15860594388E+00 .71E-11 -.65112743966E-01 -.86170430057E-03 -.93502518785E+00 -.61731643844E-02
38 -.11254651941E+00 -.48E-11 .90506809736E-01 -.92708550033E-03 .86399706108E+00 -.54812308244E-02
39 .16161236955E+00 .71E-11 -.64874505062E-01 .16604680248E-02 -.98646696093E+00 .71208634732E-02
40 -.11600476054E+00 -.47E-11 .80000000002E-01 -.22173379049E-02 .89901711612E+00 .61300314700E-02

```

Table 3.1 Normalized coordinates at the intersections of the refined QPO

Table 3.2 Time vectors with the epochs of intersections of the refined QPQ

SPCSM = .10000000000E-01 IPA = 1 1 11 1 1 1 1 1 1 1 1 1 0 0 0 ICD = 2
 ZO = .080000 T0= 16587.00000 NIT = 40 NHR = 1 IBETA = 1
 BOUND1 = .2000E-08 BOUND2 = .3000E-09 NUMITM = 12 EPS = .1000E-05

VECTOR TS (normalized time)

-.28914675855E+02	-.27249976526E+02	-.25725649263E+02	-.24233673082E+02	-.22702943362E+02	-.21135813058E+02
-.19605457567E+02	-.18113754547E+02	.16589230332E+02	-.15022596634E+02	-.13406096560E+02	-.11993584104E+02
-.10475312405E+02	-.89101529894E+01	.73675809736E+01	-.58731096759E+01	-.43606367621E+01	-.27979404526E+01
-.12497033589E+01	.24745095331E+00	.17546158045E+01	.33140539968E+01	.48674935968E+01	.63692206643E+01
.78707272214E+01	.94260777252E+01	.10983921244E+02	.12489153316E+02	.13997714326E+02	.15538202655E+02
.17099717800E+02	.18610076483E+02	.20105502799E+02	.21650552212E+02	.23214889606E+02	.24730968285E+02
.26224257154E+02	.27763417200E+02	.29329508167E+02	.30851624005E+02	.32343246656E+02	

VECTOR TS (in mod. jul. days (1950.0))

.16597000000E+05	.16677956129E+05	.16766565426E+05	.16953294151E+05	.16942201436E+05	.17033373065E+05
.17122332704E+05	.17209045630E+05	.17297666375E+05	.17388734950E+05	.17478051853E+05	.17564811752E+05
.17653069038E+05	.17744051912E+05	.17833721778E+05	.17920595545E+05	.18008515747E+05	.18099355441E+05
.18189354618E+05	.18276384349E+05	.18363995993E+05	.19454646292E+05	.18544947891E+05	.18632185306E+05
.18719526163E+05	.18809938844E+05	.18900496445E+05	.18987995736E+05	.19075107238E+05	.19165237280E+05
.19256008312E+05	.19343805614E+05	.19430734096E+05	.19520548773E+05	.19611483863E+05	.19699613669E+05
.19786419701E+05	.19875890229E+05	.19966927254E+05	.20055408000E+05	.20142116175E+05	

VECTOR TS (calendar dates)

1.00-JUN-1995	30.96-AGO-1995	27.57-NOV-1995	22.29-FEB-1996	21.28-MAI-1996	20.37-AGO-1996
17.33-NOV-1996	12.05-FEB-1997	11.67-MAI-1997	10.73-AGO-1997	9.05-NOV-1997	2.81-FEB-1998
2.07-MAI-1999	1.05-AGO-1990	29.72-OCT-1998	24.60-GEN-1999	22.52-ABR-1999	22.36-JUL-1999
20.35-OCT-1999	15.32-GEN-2000	12.00-ABR-2000	11.65-JUL-2000	9.95-OCT-2000	5.19-GEN-2001
2.53-ABR-2001	1.94-JUL-2001	30.50-SET-2001	27.00-DES-2001	24.11-MAR-2002	22.24-JUN-2002
21.01-SET-2002	17.81-DES-2002	14.73-MAR-2003	12.55-JUN-2003	11.48-SET-2003	8.61-DES-2003
4.42-MAR-2004	1.89-JUN-2004	31.93-AGO-2004	28.41-NOV-2004	23.12-FER-2005	

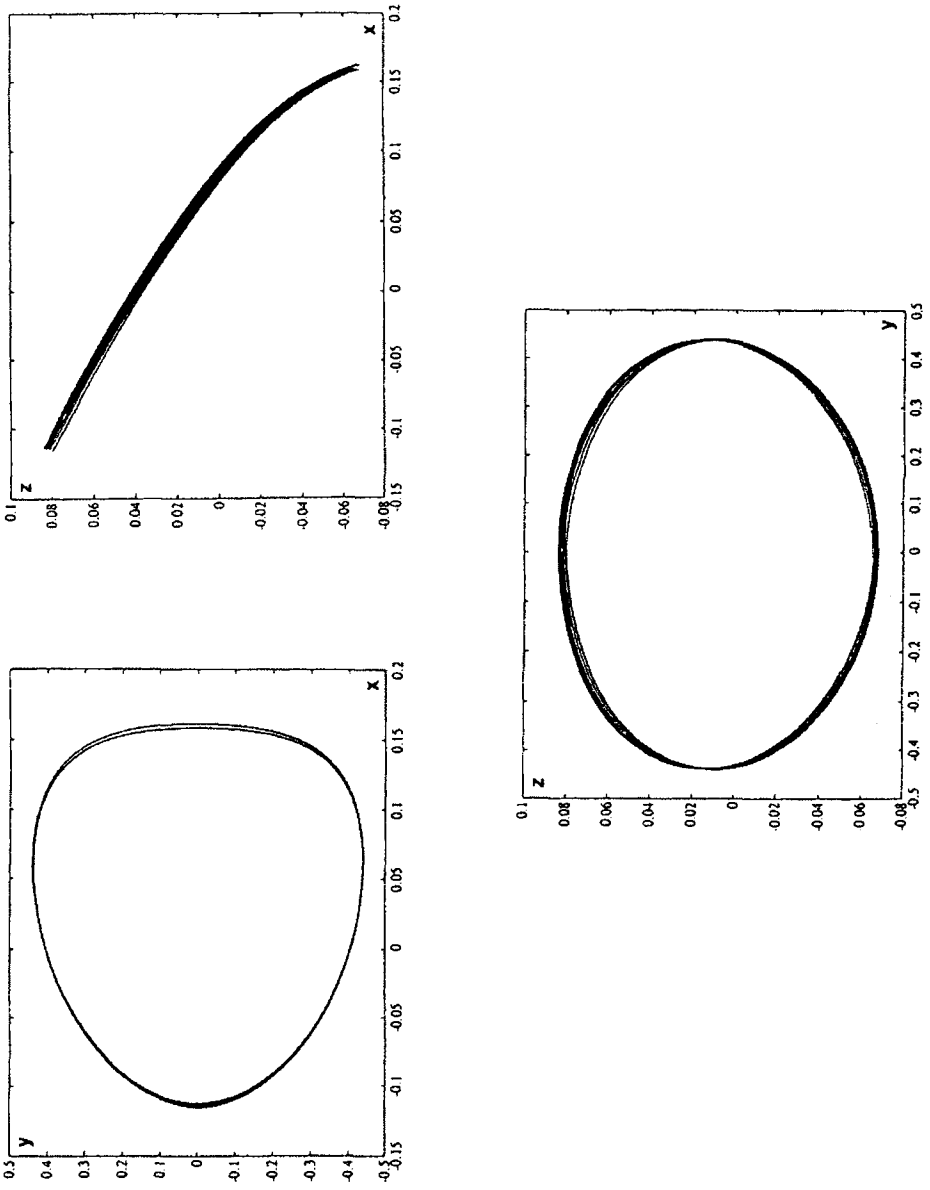


Fig. 3.1 Projections of the refined quasi-periodic halo orbit.

3.3 The Equations of Motion for the Simulations of the Control

The purpose of the next sections of this chapter is the discussion of the numerical results obtained for the control of an spacecraft in the vicinity of a halo orbit in the Earth+Moon-Sun system. The methodology used for the definition of the control strategy is the same one that was employed in a previous work [2]. The only difference is that now all the computations have been done using the JPL ephemeris for the computation of the motion of the bodies of the solar system. Nevertheless the main ideas of the method have been included in the text.

The on/off controls are supposed to act in a very short time. Then, when a control is applied we merely shift the velocities by the suitable amount.

The simulation program is suited to this approach because we have used a one step method (a Runge-Kutta-Fehlberg algorithm of order 7/8 and automatic step size control).

The mass of the Sun has been altered, as explained in Appendix B, to account for the radiation pressure effects.

As the effects of the radiation pressure have some uncertainties, we have added a random contribution to the acceleration given above. In order to avoid conflict with the numerical integration scheme, the random forces have been kept fixed at each step of the integration.

The "standard" random errors in the radiation pressure consist in a relative error, in the modulus of the radiation pressure acceleration, which is supposed distributed according to a centered normal law with typical deviation of 5%. Normally distributed errors with zero mean and typical deviation equal to 0.05 rad. have been used for both ecliptic latitude and longitude of the radiation pressure acceleration.

These standard values can be modified using an amplification (or reduction if less than 1) factor.

A systematic noise can be introduced in the following way. The nominal orbit is computed with a given value of s/m (spacecraft section facing to the Sun / mass). If the simulation is done with a value of s/m greater (less) than the nominal one, a net acceleration outwards (towards) the Sun will result.

In all the computations the adopted value of s/m has been 0.01. This is pessimistic because, probably, the real figure will be 1/3 of that one.

3.4 The Effect of Errors

The current point will be known, in the applications, through tracking. As specified by the Agency, for the L_1 case in the Earth+Moon-Sun problem the tracking errors have been supposed to be distributed according to centered normal laws with 2σ deviations equal to 3, 5 and 30 km, 2, 2 and 6 mm/sec in $x, y, z, \dot{x}, \dot{y}, \dot{z}$, respectively. Here the coordinates refer to the adimensional system.

When the simulation is started, at a given epoch, the nominal point is computed.

Then, random variations in the 6 coordinates are introduced due to tracking errors according to the given laws. The integration of the motion of the spacecraft in the solar system field is done with the random errors specified in the preceding section.

At every step the unstable component is computed, but to this computation (not to the current point) again tracking errors are applied. Hence, for a given epoch we have 3 different points: The nominal one, the real one and the estimated point.

When a too large unstable component forces to the execution of a manoeuvre the x and y components of the jump in velocity are computed. Then, again some execution errors are introduced. They can be due to a wrong direction for the manoeuvre or wrong modulus of the same.

The “standard” assumptions on those errors are the following. We have supposed that, if an (x, y) manoeuvre should be done, the x and y components are modified with a centered normal deviation with 2σ value equal to 5% of the modulus of the scheduled manoeuvre. For the z component (theoretically zero) we have used a 2% of the same amount. If (x) only or (y) only manoeuvres are done it is supposed, also, that the errors are of 5% in the active component and 2% in the nonactive ones.

Both the standard values of the tracking and the execution of the manoeuvre errors, can be modified using amplifying factors.

3.5 When a Control is Applied

It is clear that to do a manoeuvre when the unstable component is too small is nonsense. The small instability can be due to tracking errors. In Chapter III of [2] it was found that, taking a box, centered at the origin, with 2σ half sides in each one of the six variables, the maximal unstable component, changing from one point in the orbit to another one, is near 2×10^{-7} in adimensional units. Furthermore, the random errors in the acceleration should cancel on the average, but locally can have some importance. With the used values their amount can be estimated in 1×10^{-7} , approximately, in the unstable component (u.c. for shortness). Hence, manoeuvres with an u.c. less than 2, 3 or 4 times 10^{-7} should not be done.

A value is given by the user of the simulation program for this lower bound of the u.c. in modulus. Below this value no manoeuvre should be done.

When the u.c. is greater than the given value a manoeuvre can be useful. However, too frequent manoeuvres should be avoided because to settle down the tracking errors, some time interval is necessary. Typically one month, at least, has been asked, between two consecutive manoeuvres.

If the u.c. is greater than the lower bound and the time since last control is big enough we must be sure that the u.c. is increasing in an exponential way. If this is not so, the deviations can be due to the presence of small oscillations in the motion around the nominal orbit, as happens when an analytic one is taken. If the u.c. increases by a factor λ_1 (the dominant eigenvalue) in one period, T , in

a step of integration, Δt , should increase, in theory by a factor $\exp(\Delta t \frac{\ln \lambda_1}{T})$. A value m , slightly smaller than $\frac{\ln \lambda_1}{T}$ has been used. If for three consecutive steps of integration the quotient of successive u.c. is greater than $\exp(m\Delta t)$, then a clear exponential behavior is present. In any case an upper bound of the u.c. must be given such that, if the u.c. is greater than this upper bound, a manoeuvre is done unconditionally.

From the unstable character of the behavior it follows that the best choice (from the optimal point of view, i.e. the one such that the foreseen fuel used divided by the time interval between two consecutive manoeuvres reaches a minimum) is to perform manoeuvres at a value of the u.c. roughly equal to $e = \exp(1)$ times the sum of the total effect of the diverse errors. Using a lower bound 0.4×10^{-6} , a value of the upper bound of 10^{-6} is suitable.

3.6 Magnitudes Related to the Control

The control algorithm requires several magnitudes mainly related to the behavior of the unstable manifold. These magnitudes, as well as a suitable form of the nominal orbit, are computed in a program independent of the one which performs the simulations of the control.

First of all program NUNOPF computes, for given fixed time intervals, the normalized coordinates of the points of the qpo. It also computes the values of the projection factors related to the unstable manifold from the eigendirections given by routine SDIREC. In fact this last routine computes for the initial points in each revolution the unstable and stable directions as well as a basis for the remaining four directions related to the central part. These computations are performed in such a way that under the differential of the flow, from one revolution to the next one the directions are mapped in a continuous way. This is accomplished in the following way:

Assume that the qpo is split in N revolutions. Associated to the i th revolution we have the variational matrix A_i in normalized coordinates. The individual use of each one of these matrices in order to get the above directions implies a certain discontinuity when we change from one revolution to another. Nevertheless this was the approach used in [2]. In order to avoid this discontinuity it is better to use the composed matrix over the whole revolutions.

This implies some technical difficulties, for example:

Due to the large value of the unstable eigenvector of each one of the matrices A_i (roughly 1700) it is not possible to perform a direct computation of the eigenvalues of the matrix $A = A_N \times A_{N-1} \times \dots \times A_1$ because of the possible overflows during the computation of A as well as the big rounding errors. We must have into account that the dominant eigenvalue of A is of the order of 1700^N .

To avoid this difficulty we have proceeded as follows. For the computation of the unstable and the stable eigenvectors we have used the power method applied to

the matrix A and its inverse, respectively. For the remaining ones the procedure is a little bit more tricky. We start by taking four linearly independent vectors.

Assume that u_0 is the unstable vector of A . Let

$$\bar{u}_i = A_i \times A_{i-1} \times \dots \times A_1 u_0$$

and $u_i = \frac{\bar{u}_i}{|\bar{u}_i|}$.

We define $\delta_i = |A_i u_{i-1}|$. We note that

$$u_i = \frac{A_i u_{i-1}}{\delta_i}.$$

Let w_0^0 be some arbitrary vector. We introduce $\bar{w}_1^0 = A_1 w_0^0$. The initial vector w_0^0 is modified in the following way:

$$w_0^1 = w_0^0 - \frac{\lambda_1}{\delta_1} u_0,$$

where $\lambda_1 = (\bar{w}_1^0, u_1)$. Note that w_0^1 has no unstable component under A_1 . This is the first step of the procedure. In the second step we take $w_1^0 = \bar{w}_1^0 - \lambda_1 u_1$. That is: $w_1^0 = A_1 w_0^0$. Now we can compute $\bar{w}_2^0 = A_2 w_1^0$. The vector w_1^0 is modified in the following way:

$$w_1^1 = w_1^0 - \frac{\lambda_2}{\delta_2} u_1,$$

where $\lambda_2 = (\bar{w}_2^0, u_2)$. Note that w_1^1 has no unstable component under A_2 . Then we perform another correction in the following way:

$$w_0^2 = w_0^1 - \frac{\lambda_2}{\delta_1 \delta_2} u_0,$$

Note that w_0^2 has no unstable component under $A_2 A_1$.

Iterating the procedure till obtaining w_0^n , we arrive to a vector without unstable component under A . If this is done with four linearly independent vectors we get a basis of the central part.

It must be noted that the stable components have not been taken into account so the procedure works only for large values of N , say for instance $N \geq 5$ in the problem considered.

3.7 Description of the Program

The program CONSIM developed in [2] has been adapted to use the JPL numerical ephemeris, in the way that is explained in Appendix A, in order to perform the simulation of the control according to what it has been said in the preceding sections.

For the nominal orbit and for the projection factors several possibilities are offered:

- (1) Numerical halo orbit (in this case the projection factors on the Floquet modes 3 and 4 are also available optionally).
- (2) Analytically quasi-periodic orbit (generated by routine ANAQPO from the output of programs ANACOM and QPO) and analytic parameters of control (it is enough to use the ones of the RTBP) generated by program CONA from the output of ANAVAR. See [2] for the above programs.
- (3) Refined numerical quasi-periodic orbit and related normalized projection factors. They are generated by programs GQTPS and NUNOPF.

To perform checks, a nominal orbit of any kind can be used with projection factors of any kind. To allow for this possibility several additional parameters can be requested.

The determination of a point in the orbit can be done in the following ways:

- (1) The point in the nominal orbit is computed at the current time.
- (2) It is computed at a time near the current one such that the distance from the actually computed (by integration) point to the selected point in the nominal path be a local minimum. This is done writing

$$d^2 = (x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2,$$

where x_n, y_n, z_n are the nominal values and x, y, z the real ones. Then the nominal ones are considered functions of time and the other ones are constant. By time differentiation

$$\frac{1}{2}(d^2)' = (x - x_n)\dot{x}_n + (y - y_n)\dot{y}_n + (z - z_n)\dot{z}_n = 0.$$

This equation is solved for t using Newton's method.

The nominal orbit is computed by routine NOMORB. The type of the manoeuvres can also be selected:

- (1) Full optimal control ((x, y) -plane manoeuvres).
- (2) x -axis manoeuvres.
- (3) y -axis manoeuvres.
- (4) Radiation pressure manoeuvres.

The random generator used to obtain normally distributed points (for the noise in acceleration and also for tracking and manoeuvres errors) is called a given (by the user) number of times before starting the proper computations. This allows to do statistics with all the other data fixed.

Each time a manoeuvre is done, written output is produced related to it. At the end of the run a statistics of the manoeuvres is given. The available information is:

- Number of manoeuvres.

- Total amount of Δv required (in adimensional and physical units).
- Minimum and maximum observed time between two consecutive manoeuvres (in Julian days).
- Maximal and minimal manoeuvre (in adimensional and physical units).

The normalized projection factors are computed by the routine `MAGCON`, and from them the unitary controls can be computed in any case.

The numerical RTBP values are read from the output of program `PAPUS`. They are given as Fourier series and the only thing to do is the evaluation of the series at the given angle.

The analytic values are obtained from the output of `CONA`. From this output, and with the x and z -amplitudes determined when the nominal orbit is computed, a Fourier series with numerical coefficients is obtained for the projection factors, determinant and first Floquet mode. From a given angle (note that the period should be given if the orbit chosen is the numerical one) the three magnitudes can be determined and from them the normalized projection factors.

Finally, if numerically refined projection factors are used, the only thing to do is to obtain them at a given epoch using the Lagrange interpolation routine `LAG`.

3.8 Numerical Results

In the Table of the next page we present a summary of the results obtained for the refined halo orbit of amplitude $z_0 = 0.08$ and initial epoch $t_0 = 16\,587$ MJD (since 1950.0). In all the simulations the (x, y) control has been used. The time interval used in the simulations is of roughly 9 years, and goes from 16\,677.0 MJD to 19\,966.0 MJD (we have skipped the first half revolution and the last revolution of the refined halo orbit).

The results are not sensitive to the value of β (within moderate values) and are quite similar to the ones obtained in [2] when an analytic model of solar system was used.

A first look to the Table shows that, for the standard values of the different errors, at most 18.6 cm/sec per year are required. The time interval between manoeuvres ranges between 1.5 and 5 months. This is changed if the standard errors are altered. In the most pessimistic case (errors 2, 2 and 4 times the standard ones) at most 28 cm/sec per year are required.

All the computations have been done with the option: minimal distance from the nominal to the current point. In the interactive runs of `CONSIM` additional information appears which allows to check the values of the changes of the nominal point in time. The runs rarely exceed 2^m in runs covering 9 years controlled by on/off algorithm (on a HP 9000/835S).

In the table the following notation has been used:

$W_{max,min}^u$ refer to the upper and lower bounds for the unstable component.

Δt : to the desired minimum time interval between manoeuvres.

$a_i, i=1, 2, 3$: to the amplifying factors of errors in tracking, execution of manoeuvres and radiation pressure.

calls: to the number of previous calls to the random number generator.

man: to the number of manoeuvres done.

Δv : to the total amount of increment of velocity required for the time span of 9 years considered (cm/sec).

$t_{min,max}$: to the minimum and maximum experimental values of the time interval between manoeuvres (jd).

$v_{min,max}$: to the minimum and maximum values of the manoeuvres (cm/sec).

W_{max}^u	W_{min}^u	Δt	a_1	a_2	a_3	$calls$	man	Δv	t_{min}	t_{max}	v_{min}	v_{max}
1.D-6	.4D-6	60	1	1	1	100	40	161	47	150	1.8	5.9
1.D-6	.4D-6	60	1	1	1	400	41	170	45	149	2.1	5.9
1.D-6	.4D-6	60	1	1	1	700	41	163	50	150	2.0	6.3
2.D-6	.8D-6	60	1	1	1	101	34	233	60	264	4.4	11.9
2.D-6	.8D-6	60	1	1	1	401	36	230	63	165	3.8	11.4
2.D-6	.8D-6	60	1	1	1	701	37	232	62	161	3.8	10.8
.5D-6	.2D-6	60	1	1	1	102	47	106	31	117	0.8	3.0
.5D-6	.2D-6	60	1	1	1	402	47	106	31	117	0.8	3.0
.5D-6	.2D-6	60	1	1	1	702	43	98	39	129	0.9	2.9
1.D-6	.4D-6	30	1	1	1	103	40	156	46	167	2.0	5.9
1.D-6	.4D-6	30	1	1	1	403	38	140	38	177	1.9	5.6
1.D-6	.4D-6	30	1	1	1	703	40	152	42	175	2.0	5.6
1.D-6	.4D-6	60	2	1	1	104	46	197	36	149	1.6	6.0
1.D-6	.4D-6	60	2	1	1	404	40	172	56	150	1.7	6.6
1.D-6	.4D-6	60	2	1	1	704	48	213	30	148	2.1	6.4
1.D-6	.4D-6	60	4	1	1	105	65	326	20	101	2.2	7.0
1.D-6	.4D-6	60	4	1	1	405	60	303	11	156	1.6	6.8
1.D-6	.4D-6	60	4	1	1	705	49	229	33	176	1.8	6.1
1.D-6	.4D-6	60	.5	1	1	106	40	150	60	139	2.0	5.4
1.D-6	.4D-6	60	.5	1	1	406	37	135	60	193	2.0	5.5
1.D-6	.4D-6	60	.5	1	1	706	39	136	54	183	2.0	5.7
1.D-6	.4D-6	60	1	2	1	107	42	177	42	163	2.2	6.3
1.D-6	.4D-6	60	1	2	1	407	41	180	43	159	2.2	6.3
1.D-6	.4D-6	60	1	2	1	707	40	173	38	185	2.1	6.1
1.D-6	.4D-6	60	1	4	1	108	49	231	32	209	2.1	6.8
1.D-6	.4D-6	60	1	4	1	408	47	211	29	133	1.7	6.5
1.D-6	.4D-6	60	1	4	1	708	47	205	33	141	2.2	6.1
1.D-6	.4D-6	60	1	.5	1	109	39	151	54	169	1.9	5.9
1.D-6	.4D-6	60	1	.5	1	409	40	153	57	167	2.0	5.9
1.D-6	.4D-6	60	1	.5	1	709	37	149	59	142	1.9	6.0
1.D-6	.4D-6	60	1	1	2	110	42	184	46	137	2.1	6.2
1.D-6	.4D-6	60	1	1	2	410	41	169	47	169	2.0	6.0
1.D-6	.4D-6	60	1	1	2	710	38	162	50	148	1.8	5.8
1.D-6	.4D-6	60	1	1	4	111	54	245	27	124	2.7	6.3
1.D-6	.4D-6	60	1	1	4	411	53	240	34	103	1.9	6.2
1.D-6	.4D-6	60	1	1	4	711	51	229	35	139	2.1	5.8
1.D-6	.4D-6	60	1	1	.5	112	41	157	61	123	1.8	5.8
1.D-6	.4D-6	60	1	1	.5	412	37	153	47	135	1.9	6.1
1.D-6	.4D-6	60	1	1	.5	712	35	142	52	167	2.2	6.0
1.D-6	.4D-6	60	2	2	4	713	54	251	24	112	2.4	7.0

Furthermore keeping the standard values fixed (that is, $W_{max}^u = 1.D-6$, $W_{min}^u = 0.4D-6$, $\Delta t = 60$, $a_1 = a_2 = a_3 = 1$) we have done 999 simulations. The Table 3.4 shows the relevant statistics.

	man	Δv	t_{min}	t_{max}	v_{min}	v_{max}
mean	39.41	159.13	51.98	164.96	2.04	5.93
st. dev.	2.09	11.82	5.43	30.00	0.20	0.21
minimum	32.00	115.34	25.46	110.71	1.51	5.22
maximum	46.00	196.91	62.34	282.08	3.02	6.57

Table 3.4

3.8.1 Discussion on the Effect of the Different Errors and Magnitudes

The standard values of W_{max}^u and W_{min}^u in the runs have been 10^{-6} and 0.4×10^{-6} . An increase in these values is not suitable. A decrease is possible without producing too near manoeuvres. This produces a reduction on Δv of roughly 35% if $W_{max,min}^u$ are halved.

The comparison of results for different values of a_1 shows that the tracking errors are responsible of the most important part of the total Δv required. Hence, it is important to reduce as much as possible the tracking errors.

The errors in the execution of the manoeuvres have a smaller contribution than the ones due to tracking and the ones which have the smallest contribution are those coming from the random errors in the radiation pressure.

If the contribution of all these three errors is set equal to zero, there is still some small amount of Δv required (45 cm/s roughly). This is due to the error propagation during the integration.

3.8.2 Conclusions

The station keeping of quasi-periodic halo orbits, for the Earth+Moon–Sun system, is not a serious problem in what concerns the Δv requirement. A dramatic reduction of the fuel consumption is achieved if the nominal orbit is a true orbit of the motion of a spacecraft in the Solar System. The tracking errors give the most important contribution to Δv .

The problem becomes much more critical when the time scales change. This happens, for instance, for the Earth–Moon system. The methods applied in this chapter can be used for this problem, of course, but the time scale is much shorter. This means that, using the same figures as above for the “standard” errors in tracking, execution of manoeuvres and model, the total Δv per year will be increased by a factor around 13 and this is still feasible. However the reduction of the time interval between manoeuvres requires a fast and accurate tracking.

This page is intentionally left blank

Chapter 4

Transfer From the Earth to a Halo Orbit

4.1 Introduction

The usual way to obtain the transfer trajectories from the Earth to a halo orbit is by means of an optimization procedure. The procedure looks for an orbit between the Earth and the halo maintaining some boundary conditions and minimizing the total fuel to be spent in manoeuvres during the transfer (see [5]). The approach we propose is a geometrical one, avoiding the main procedure of optimization which gives no information about the characteristics of the problem.

Due to the strong hyperbolic character of the halo orbits, the stable manifold approaches the halo orbit in a very fast way. This fact means that if we are able to put the satellite in the stable manifold of a halo orbit, it will be close to the halo orbit (say at few km) in a reasonable period of time.

Proceeding in this way we shall avoid the insertion manoeuvre into the halo orbit. So we must take into account only the insertion manoeuvre into the stable manifold and the tracking manoeuvres, in order to follow the transfer path, which of course will be needed, in any case, because we are approaching to an hyperbolic orbit.

Following this procedure, the main thing we have to look for are the approaches of the stable manifolds of the halo orbits to the Earth, since the insertion manoeuvres into the stable manifold should be done, in principle, at the moment of the departure of the satellite from the vicinity of the Earth. If this were not the case, then we had to do two manoeuvres: at the departure from the Earth and for the insertion in the stable manifold. If the last one is done far away from the Earth it should be of a small magnitude. It is difficult to predict beforehand the regions where this manoeuvre could be accomplished.

For convenience, the proposed method has been split and implemented in several parts. In this way, when necessary, the user can go backwards without doing all the computations from the starting point.

The steps of the method are:

- To obtain the quasi-periodic halo orbits in the solar system to be explored (program GQPTPS). This point is described in Chapter 3.
- To obtain the local approximation of the stable manifold near the quasi-periodic halo orbits selected (program VECSWS).
- To globalize, by numerical integration, the stable manifold from the local approximation obtained (program ESTWS1).
- To explore the manifold obtained, searching the passages near the Earth (program SELWS).
- To obtain, from the previous search, the ranges of the manifold well suited for the transfer (programs AFMIN and INTER).
- To study the characteristics of the orbits contained in the ranges of the previous point (program FINES).

4.2 Local Approximation of the Stable Manifold

What we are going to do is to compute the linear approximation of the stable manifold for the quasi-periodic halo orbit.

We know that if an orbit is periodic, the eigenvectors of the variational matrix computed over one period of the orbit, that is the monodromy matrix, give the directions of the tangent spaces to the invariant manifolds at the starting point. Seen in the normalized adimensional reference frame our orbit is not periodic but quasi-periodic with small deviations from the halo periodic orbit that we would see in the restricted three-body problem. So, in this reference system, the eigenvector associated to the eigenvalue with smallest absolute value of the variational matrix computed over a revolution (two successive passages through the plane $y = 0$ in adimensional coordinates), approximates quite well the stable direction when the orbit crosses $y = 0$.

We want to use the orbits computed in Chapter 3 as nominal orbits for a mission lasting a certain time span. In what concerns the transfer, we are interested in the stable manifold associated to the initial revolutions. Due to possible small bends of the manifold because of the boundary conditions in the computations of the qpo by the parallel shooting procedure, the first revolution is skipped and we shall compute the stable manifold from the beginning of the second one*. From this point we shall compute the stable direction in a set of points of the halo orbit usually equally spaced in the whole revolution† in the following way.

As in the computations of the magnitudes related to the control (Chapter 3, Section 7), instead of computing the stable direction associated to the selected revolution, by means of its variational matrix, in order to avoid the small discontinuity

*This fact corresponds in the programs to take $NV0 = 3$, which means to start the computation of the stable manifold in the third cut of the orbit with $y = 0$.

†In our computations the usual number which was taken was 250.

obtained in the stable direction when we go from one revolution to the next one, the variational matrix \bar{A} associated to the whole quasi-periodic orbit has been taken as a monodromy matrix. We recall that \bar{A} is a 7×7 matrix because the time was added in the equations of motion in order to get an autonomous system and its pattern is:

$$\bar{A} = \left(\begin{array}{c|cccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline g_1 & & & & & & \\ g_2 & & & & & & \\ g_3 & & A & & & & \\ g_4 & & & & & & \\ g_5 & & & & & & \\ g_6 & & & & & & \end{array} \right),$$

where A is the part of \bar{A} associated to positions and velocities. We note that if v is an eigenvector of A , then $(0, v)$ is an eigenvector of \bar{A} . We shall compute the stable direction for the 6×6 matrix A .

If A_i is the 6×6 matrix related to positions and velocities of a certain variational matrix \bar{A}_i associated to the i th revolution of a numerical qpo made of N revolutions, we have:

$$A = A_N \times A_{N-1} \times \cdots \times A_1.$$

Because of possible overflows and rounding errors, due to the fact that the eigenvalue of each A_i associated to the unstable manifold is of the order of 1700, the above product must not be computed. With the power method applied to

$$A^{-1} = A_1^{-1} \times A_2^{-1} \times \cdots \times A_N^{-1},$$

we can compute the eigenvector v_1 of the matrix A associated to the stable manifold. Then the eigendirections v_j related to the beginning of each j th revolution are computed by means of:

$$v_j = \frac{A_j^{-1} v_{j+1}}{|A_j^{-1} v_{j+1}|}, \quad j = N, \dots, 2,$$

where $v_{N+1} = v_1$. The "contraction"[†] corresponding to the stable direction associated to the j th revolution is:

$$\delta_j = |A_j^{-1} v_{j+1}|^{-1}.$$

When we have the direction of the stable manifold at the beginning of a revolution, we can obtain the stable direction at any intermediate point in the revolution

[†]Eigenvalue in the case of a periodic orbit

transporting this vector by means of the 6×6 differential matrix of the flow corresponding to the coordinates of position and velocity. That is, if $v_j(t_0) = v_j$ is the initial vector at time t_0 we have

$$v_j(t) = A_j(t)v_j(t_0),$$

where $A_j(t) = Id$.

Essentially this is the way by which the stable directions are computed. Given the vector $v_j(t_0)$ in normalized coordinates, corresponding to the stable direction at the initial point $p_j(t_0)$ of a certain revolution in the phase space in normalized coordinates, and using the routine DTRAN, $p_j(t_0)$ is converted into ecliptic coordinates, and the differential matrix, D_{ne} , of the transformation from normalized to ecliptic coordinates at time t_0 and point $p_j(t_0)$ is obtained. Then the stable direction in ecliptic coordinates, $v_j^e(t_0)$, at the image point, $p_j^e(t)$, is computed by means of

$$v_j^e(t_0) = D_{ne}v_j(t_0).$$

From now on the computations are performed in ecliptic coordinates. The differential $D_j(t)$ of the flow is given by the integration of the vector field given by routine DERAES, which contains also the variational equations. So we have that the stable direction in ecliptic coordinates at any point of the selected revolution can be obtained by

$$v_j^e(t) = D_j(t)v_j^e(t_0).$$

The main output of this study is $v_j^e(t)$ at some points of the qpo. It is scaled in such a way that the three components of these vectors relative to the positions have Euclidean norm equal to one. So what we write is:

$$\bar{v}_j^e(t) = \frac{v_j^e(t)}{\sqrt{(v_j^e)_x^2 + (v_j^e)_y^2 + (v_j^e)_z^2}}.$$

In this way if we want to take initial conditions at a selected physical distance from the point $p_j^e(t)$ which is in the qpo, we have only to multiply $\bar{v}_j^e(t)$ by the selected distance in km with the desired sign.

4.2.1 Main Program and Basic Routines

Letting aside the routines associated to the numerical integration and to the changes of coordinates, the main program and routines are the following:

Program VECWS It is the main program of the local computation of the stable manifold. It selects the orbit or orbits computed by program GQPTPS, and produces a file with the stable directions at the desired points of the orbit. Through the standard output writes the main incidences of the execution of the program.

Routine RGQTP This routine reads the output of program GQTPS.

Routine SDIREC It is used to compute the stable direction v_j , from the variational matrices A_1, \dots, A_N , in normalized coordinates and corresponding to each revolution of the qpo.

Routine DIRS It computes the stable directions in ecliptic as it has been explained, at selected different epochs from the beginning of the chosen revolution.

4.2.2 Output and Sample of Results

Starting with the quasi-periodic halo orbit computed in Chapter 3, stored in the file `gqtps1.dat` we present here a sample of the results obtained by program VECWS.

The data which appear in the file of results correspond to:

- The name of the file with the results of program GQTPS: `gqtps1.dat`,
- The parameters defining the model of solar system and the spacecraft section divided by the mass: `SPCSM`, `IPA`, `IC0`. For coherence, these values should be the same ones used in the computation of the QPO.
- The value which appears in the line following `SPCSM` means the following: if it is not set equal to zero, then the file contains information about the linear behavior of the stable manifold. This parameter is used by other programs that have the output generated by this one as input values. If it is set equal to one, the end of file has been reached.

- The next two lines contain general information about the computations. The quasi-periodic orbit corresponds to a z -amplitude $Z0 = 0.08$, and the initial epoch for its numerical refinement has been taken as $T0 = 16587$. $NV0 = 3$ means that the revolution that has been selected for the computation of the stable directions is the second one. The qpo has a certain number of half revolutions and in this example the first two ones have been skipped.

The second revolution lasts for 175.716... days, and the compression factor of the stable direction along this revolution is equal to 0.0005890...

The number of points, `NPT`, that have been taken along the second revolution, at which the stable directions are computed, is 250.

The time interval between two of the above mentioned points, equally spaced along the qpo, is `DDIA = 0.702864...`

- The following lines must be considered in groups of three. In the first line of each group appear the modified Julian day and the component related to the time on the stable direction, which is always equal to zero because of the use of the variational matrix for the computations.

The second line gives the position and velocity, in ecliptic coordinates (km, km/day), of the point of the qpo corresponding to the epoch that appears in the first line.

The third lines contains the coordinates of the stable direction related to

position and velocity. They are given in ecliptic coordinates and they have been scaled in such a way that the norm of the vector formed by the first three ones is one.

In the example only the first 10 groups are given, of a total of 251 (the zero must also be included).

INPUT FILE (OUTPUT OF PROGRAM QPO) = gqtps1.dat

0

Z0= .080000 T0= 16587.00000 DDIA= .702864041878E+00 NPT-250

.630637166321E+08 .132522654219E+09 .115941911399E+06 -.235455451546E+07 .110883537112E+07 -.222222101461E+03

```

.167672602899E+05      .000000000000E+00

```

1111197062929E+00 -993595020215E+00 -200638791925E-01 -176949750819E-01 -401588946710E-01 214824984546E-02

597346640664E+08 134040820955E+09 115516711674E+06 - 233161219118E+07 105101682087E+07 - 382672633871E+03

```

1.101904650550E+00      .9940015000197E+00      -.191578895005E-01      -.174491553975E-01      -.402612417372E-01      .212869490496E-02
167606740100E+05      00000000000000E+00

```

.580561320016E+08 .134769300225E+09 .115219399749E+06 -.239459011177E+07 .102184254501E+07 -.462737403899E+03
 826600232807E-01 0055205723270E+00 -181078046045E-01 -172004460254E-01 -402623402442E-01 8100180052237E-02

.926699232807E-01 .995530573270E+00 -.181978946045E-01 -.172004460254E-01 -.403623423442E-01 .210012995227E-02
1.67603163880E+05 000000000000E+00

.563686083379E+08 .135477213848E+09 .114866263719E+06 -.240719737725E+07 .992490002765E+06 -.542652646762E+03

.832525010004E-01 .996379274197E+00 -.172442133446E-01 -.169486397707E-01 -.404621370784E-01 .208655088480E-02

```
.167700797461E+05      .000000000000E+00
```

```
.737322209612E-01 .997145230685E+00 -.162772391463E-01 -.166934758989E-01 -.405606239727E-01 .206396129957E-02
```

```
.167707026101E+05      .000000000000E+00
```

.5296763343\$1E+08 .136830075977E+09 .113991415138E+06 -.243129024462E+07 .933318121620E+06 -.7018895530838+03

```
.641091195357E-01 .997825642221E+00 -.152973369986E-01 -.164347462679E-01 -.406578475478E-01 .204037112999E-02
```

```
.167714854742E+05      .000000000000E+00
```

.512547099047E+08 .137476398983E+09 .113470216845E+06 -.244277119651E+07 .903492423084E+06 -.781133175995E+03

.543834241720E-01 .998417654739E+00 -.143040901123E-01 -.161722772009E-01 -.407538902973E-01 .201579536681E-02

```

167721803302E+05      00000000000000E+00

```

495338503031E+08 138100905370E+09 112893425644E+06 - 245387219917E+07 0735161440342+06 - 860072236881E+03

445555300796E-01 998918363587E+00 = 133002071755E-01 = 1590582904103E-01 = 4084886341048E-01 189025200550E-02

```

.445555300796E-01   .998918363587E+00   -.133002071755E-01   -.159059290403E-01   -.408488634048E-01   .199025290550E-02
167720812022E+05     000000000000E+00

```

.167720912022E+05 .000000000000E+00

470053884343E+00 138704801030E+00 118861870482E+06 846450113717E+07 043804835303E+06 038030473000E+08

.470053224343E+08 .138704291039E+09 .112261270402E+06 -.246459113717E+07 .043394235393E+06 -.938664479001E+03
 316250700000E-01 000000000000E+00 100000000000E-01 150000000000E-01 100000000000E-01 100000000000E-01

.346259799809E-01 .999324845493E+00 -.122839200436E-01 -.156355944862E-01 -.409429000509E-01 .196376551166E-02

Local Approximation of the Stable Manifold

4.3 Globalization of the Manifold

When the output of program VECSWS is available, the next thing we have to do is the globalization of the stable manifold. This manifold will be computed from the points and directions of the output of VECSWS, selecting all of them or only a subset.

Given a displacement, D , in km from the selected point in the qpo, in the sense of the stable manifold which gives approaches to the Earth, and taking into account that the stable directions in the output of VECSWS are in ecliptic coordinates scaled as explained before, initial conditions X_{ws}^0 in ecliptic coordinates, in the linear approximation of the manifold are given by means of:

$$X_{ws}^0 = X_{qpo} + D \cdot V_{ws},$$

where X_{qpo} is the selected point of the qpo and V_{ws} is the scaled stable direction in the point X_{qpo} . All of them in ecliptic coordinates.

It must be noted that the magnitude D can not be too small, in absolute value, in order to prevent rounding errors and large integration periods of time when globalizing the manifold. However, it can not be too large because the linear approximation is good near the point X_{qpo} . Several tests about this value have been done. In spite of the fact that the attractive character of the manifold towards the qpo was significative even for large values of D , taking also into account the integration time during the globalization of the manifold, we suggest values of about 200 or 250 km (in the right sense). In our computations we took $D = 200$. This implies that when we say that we reach the qpo we mean that we are at 200 km of the nominal point X_{qpo} in the qpo. We recall that to reach the qpo has no meaning because of the asymptotic behavior of the manifold towards the orbit.

When X_{ws}^0 is computed, what we must do only is to integrate this initial condition backwards in time till a local minimum near the Earth is obtained. However, mainly due to collisions with the Moon and to the big size of the manifold, the algorithm must take into account several problems which can appear during the integration.

Concerning the orbits which pass near the Earth, in which we are interested on, the integration is stopped when we reach a distance of less than 6 000 km from the center of the Earth. We could use the radius of the Earth as stopping criteria, but a little bit less is better for the subsequent programs for the transfer.

If the orbit does not collide with the Earth, a Newton procedure is used in order to get the minimum distance to the Earth looking for a zero of the function

$$F(t) = \vec{r}_E(t) \cdot \dot{\vec{r}}_E(t),$$

where $\vec{r}_E(t)$ is the vector position of the satellite with respect to the Earth at the epoch t . If the minimum is close to the Earth (we took less than 100 000 km in our computations), and during the numerical integration the distance from the orbit to the Moon is less than some amount (100 000 km in our computations), another

Newton procedure is started in order to get the minimum distance to the Moon. Its starting point is the value obtained during the integration.

4.3.1 *Main Program and Basic Routines*

The main program and basic routines involved in the globalization of the stable manifold are the following:

Program ESTWS1 It is the main program of the globalization. It selects the orbits and the points in each orbit to be studied from the output of program VECSWS. When computing the globalization writes the results in a file, and through the standard output writes the main incidences of the execution of the program.

Routine CIWSDS This routine performs the computation of the initial conditions in the stable manifold using the formula of the previous section.

Routine MDTWS Is the main routine of the program. From the initial conditions obtained by routine CIWSDS it integrates backwards in time until it reaches the minimum distance from the orbit to the Earth. It computes also the distance from the orbit to the Moon and epochs and coordinates of both events.

Routine DTTLS This routine computes the necessary things to perform the Newton algorithm for computing the minimum distance from the orbit to the Earth.

4.3.2 *Output and Sample of Results*

With the results of program VECSWS, the program ESTWS1 produces the following output:

- The name of the file with the results of program VECSWS: `vecsws1.dat`,
- The parameters defining the model of solar system and the spacecraft section divided by the mass: `SPCSM`, `IPA`, `IC0`. For coherence, these values should be the same ones used in the previous computations.
- The value which appears in the line following `SPCSM` has the same meaning as in 4.1.2.
- The next line contains general information about the computations. The quasi-periodic orbit corresponds to a z -amplitude $Z0 = 0.08$, and the initial epoch for its numerical refinement has been taken as $T0 = 16587$. As in program VECSWS, $NV0 = 3$ means that the revolution that has been selected for the computations is the second one. $NC1$, $NC2$ denote the range of points on the qpo that shall be studied, shifting the initial conditions along the stable direction 200 km, $DESP = 200$.

- The following lines must be considered in groups of four, numbered from NC1 to NC2.

In the first line it appears an error flag of the routine MDTWS, and three epochs, given in MJD, corresponding to the moment of departure from the qpo (the integration is performed backwards), and the times of minimum distance to the Moon and to the Earth, respectively. The two last values of this line give the minimum distances from the spacecraft to the Moon and to the Earth (both in km).

In each one of the remaining three lines ecliptic positions and velocities are given. They correspond, respectively, to the three epochs given in the first one.

In the example only the first 5 groups are given.

Table 4.2 Partial output of program ESTWS1

```

OUTPUT OF PROGRAM ESTWS1
INPUT FILE (OUTPUT OF PROGRAM VECSWS): vecswws1.dat
SPCSM= .10000000000E-01 IPA- 1 1 11 1 1 1 1 1 1 1 1 1 0 0 0 IC0= 2
0
ZO= .080000 TO= 16587.00000 NVO= 3 NC1= 0 NC2=250 DESP= 200.000
0 0 .167665654259E+05 .165610252120E+05 .165604798876E+05 .407347396154E+06 .145990050217E+05
0 .630637406937E+08 .132522052722E+09 .115937716294E+06 -.235455810311E+07 .110802736003E+07 -.221788739390E+03
0 -.107931392334E+09 -.104401642190E+09 .181145567204E+05 .188083070755E+07 -.169836045257E+07 .124614589612E+05
0 -.109012717239E+09 -.103475397323E+09 .147057034323E+05 .232031521006E+07 -.211307259545E+07 -.734691947932E+05
1 0 .167672682899E+05 .165611672355E+05 .165611672355E+05 .410042325654E+06 .139450194394E+05
1 .614039671319E+08 .133292096094E+09 .115753493300E+06 -.236826967120E+07 .108000301467E+07 -.302061403899E+03
1 -.107007801592E+09 -.104740605000E+09 .149995487248E+05 .235474039243E+07 -.210109241741E+07 -.710360237826E+05
1 -.107907001592E+09 -.104748605000E+09 .149995497248E+05 .235474039243E+07 -.210109241741E+07 -.710360237826E+05
2 0 .167679711540E+05 .165618476330E+05 .165618476330E+05 .405631508432E+06 .133491319593E+05
2 .597346044633E+08 .134041019878E+09 .115512004096E+06 -.238161568101E+07 .105100876862E+07 -.392246e94990E+03
2 -.106600709951E+09 -.105994690127E+09 .151955621094E+05 .230022436690E+07 -.208843715669E+07 -.690375954218E+05
2 -.106600709951E+09 -.105994690127E+09 .151955621094E+05 .230022436690E+07 -.208843715669E+07 -.680375954218E+05
3 0 .167686740180E+05 .165625313529E+05 .165625313529E+05 .400698654207E+06 .120039601975E+05
3 .580561505356E+08 .134769499331E+09 .115215960172E+06 -.239459355186E+07 .102183447254E+07 -.462315777909E+03
3 -.105373510045E+09 -.107232432054E+09 .153759534978E+05 .242099075377E+07 -.207554116144E+07 -.643205386737E+05
3 -.105373518045E+09 -.107232432054E+09 .153759534978E+05 .242099075377E+07 -.207554116144E+07 -.643205386737E+05
4 0 .167693768820E+05 .165632120333E+05 .165632128333E+05 .395354055890E+06 .123348392758E+05
4 .563606249884E+08 .135477413124E+09 .114862814876E+06 -.240720076698E+07 .992489990338E+06 -.542235336505E+03
4 -.104136361039E+09 -.108451582960E+09 .155386084909E+05 .245246205059E+07 -.206231262460E+07 -.598204695741E+05
4 -.104136361039E+09 -.108451582960E+09 .155336084909E+05 .245246205059E+07 -.206231262460E+07 -.598204695741E+05

```

4.4 Selecting Passages Near the Earth

This step is a rough selection of the local minima which have the graphics of the minimum distances from the orbits of the stable manifold to the Earth, as computed by program ESTWS1. The objective is to give to the user a simple, but good approximation, of the behavior of those minima and at the same time, to prepare the data for new computations. An sketch of the procedure is shown in Figure 4.1.

All the local minima of the stable manifold below a certain distance (given in km) from the center of the Earth,[§] are selected.

The stable manifold is a two dimensional manifold which can be parametrized in the following way. Once the displacement D of the previous section is selected, a point X_{qpo} in the halo orbit gives an initial condition by the relation given in 4.2. Following the flow forwards or backwards we get all the points in the manifold associated to X_{qpo} . In this way X_{qpo} can be thought as one of the parameters which generate the manifold. We shall call it the *parameter along the orbit*. The other one is the elapsed time $t - t_0$ from the initial condition X_{ws}^0 at time t_0 to the point X_{ws} at time t following the flow. We call it the *parameter along the flow*.

We remark that this parameterization depends on the choice of D and the way in which the stable directions are scaled. If the stable directions are scaled in the usual way, a small change in D produces an effect equivalent to a small change in the parameter along the orbit. That is, with a small change in D we can get the same orbits of the manifold as with a small change of X_{qpo} . Only a small shift in the parameter along the flow will be observed. This is because the stable direction is transversal to the flow.

Due to the fact that the program ESTWS1 stops the numerical integration when a specific local minimum in the distance from the orbit to the Earth is found, our globalization of the stable manifold is, in fact, uniparametric. The graphic of minimum distances is a curve which depends only of the parameter along the flow.

Suppose that u_n is a certain point X_{qpo}^0 which is a minimum in the plot of minimum physical distances from the stable manifold to the Earth. Because of the discretization, u_n does not correspond to the value of the parameter along the orbit giving the exact minimum distance (it is between u_{n-1} and u_{n+1}), and later on, we will be interested on a refined value. Of course if we want to compute it, we could start again the computations of the approximation of the stable manifold near u_n like in program VECSWS, and with an iterative procedure we can obtain an improved value of the parameter along the orbit. Instead of this, it is better and easier to change the parameter along the orbit, keeping u_n and varying D . The minimum near u_n will be obtained by an iterative procedure over D . In Section 4.2, we said that to reach the qpo means to arrive at $|D|$ km of the point in which we computed the stable direction. This is not exact. However the variations in D are very small if the discretization has a reasonable number of points and, for practical purposes,

[§]In our computations we have chosen 100 000 km (see the DATA RTOL in program SELWS).

it does not matter.

The graph of the manifold can be strongly bent and with some discontinuities because of near passages and collisions with the Moon. It is necessary to obtain the interval in the parameter along the orbit, D , at which the local minimum is located, in order to prevent from possible problems in the next programs. This is done as follows.

Let D_n be the displacement for which the manifold is globalized. Let v_{n-1} , v_n and v_{n+1} be the stable directions computed at u_{n-1} , u_n and u_{n+1} respectively. According to 4.3

$$w_{n-1} = u_{n-1} + D_n v_{n-1} \text{ and } w_{n+1} = u_{n+1} + D_n v_{n+1}.$$

We know that u_{n-1} , u_n and u_{n+1} are points in the halo orbit, and the flow goes from u_{n-1} to u_{n+1} when the time increases. In order to carry them towards the D parameter we take w_{n-1} and we integrate this initial condition forward till it crosses the direction $u_n + \lambda v_n$ for a certain value of λ . The D_{n-1} is equal to the value of λ , and D_{n+1} is computed in a similar way but following the time backwards.

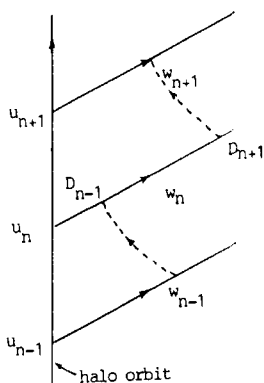


Fig. 4.1 Scheme for the estimation of the displacement along the stable manifold of the halo orbit, with a local minimum distance to the Earth.

4.4.1 Main Program and Basic Routines

The main program and routines of this section are the following:

Program SELWS It is the main program of the procedure. It reads the output from programs ESTWS1 and VECSWS and produces a file with the selected local minima and the main information concerning those minima.

Routine REST This routine is used to read the output of program ESTWS1.

Routine RVECS This routine is used to read the output of program VECSWS.

Routine MARGES Is the main routine of the program. It computes the interval in D where the local minimum is contained.

4.4.2 *Output and Sample of Results*

With the results of the preceding programs, the program SELWS produces the following output:

- Some comments related to an interactive execution of the program including some manual modification of the file (see [4]).
- The names of the files with the results of programs VECSWS and ESTWS1: `vecsws1.dat` and `estws11.dat`.
- The parameters defining the model of solar system and the spacecraft section divided by the mass: SPCSM, IPA, IC0. For coherence, these values should be the same ones used in the previous computations.
- The zero that appears under SPCSM has the same meaning as in the preceding programs.
- The next line contains general information about the computations.
The quasi-periodic orbit corresponds to a z -amplitude $Z0 = 0.08$, and the initial epoch for its numerical refinement has been taken as $T0 = 16587$.
As in the previous output files, $NV0 = 3$ means that the revolution that has been selected for the computations is the second one.
 $NC1$, $NC2$ denote the range of points of the file `estws11.dat` that shall be studied. The shift of the initial conditions along the stable direction is 200 km, as used in program ESTWS1, $DESP = 200$.
 $RTOL$ is the threshold used for the selection of the minima on the manifold.
 NTR is the number of minima that have been found.
 MAR is an index which allows to do manual modifications to the file (see [4]).
- The following lines must be considered in groups of two, numbered from 1 to NTR .

In the first line the first integer is the index of the corresponding point of the file `estws11.dat`. The next two numbers are the minimum distances to the Moon and to the Earth given in km. The remaining three calendar dates are the departure epoch from the qpo and the epochs of minimum distance to the Moon and to the Earth respectively. It can happen, and in fact in some cases it does, that these last two epochs shall be roughly the same, depending on the place where the Moon sites at the epoch of minimum distance to the Earth.

In the second line the first three numbers are the values of the parameter along the orbit, measured in the stable direction, defining the interval which contains the minimum distance to the Earth. The last three ones are the corresponding distances to the Earth.

Table 4.3 Output of program SELWS

OUTPUT OF PROGRAM SELWS

lf this file comes from program REFIN2 the variables NC2 and DESP are meaningless

lf a manual change of the displacements is done, you must change the var. MAR according to IND
of routine MDTWSQ

INPUT FILES (OUTPUT OF PROGRAMS VECSWS AND ESTWS1): vecsws1.dat, estws11.dat

SPCSM= .100000000000E-01 IPA= 1 1 11 1 1 1 1 1 1 1 1 1 0 0 0 IC0= 2

0

Z0= .080000 T0= 16587.00000 NVO= 3 NC1= 0 NC2=250 DESP= 200.000 RTOL=100000.000 NTR= 5 MAR= 0

1 10 361174.285 11080.439 4.59-DES-1995 12.23-MAI-1995 12.23-MAI-1995

1 .194300805609E+03 .200000000000E+03 .205935574332E+03 .110880286972E+05 .110804394672E+05 .111572355189E+05

2 31 84528.700 14550.662 19.35-DES-1995 27.94-MAI-1995 25.64-MAI-1995

2 .194461532603E+03 .200000000000E+03 .205759749881E+03 .177671630640E+05 .145506616514E+05 .166948398513E+05

3 72 72806.095 99156.134 17.17-GEN-1996 21.59-JUN-1995 10.55-JUN-1995

3 .194483394618E+03 .200000000000E+03 .205776310617E+03 .123080301684E+06 .991561343137E+05 .114988392661E+06

4 230 181250.325 46974.104 7.22-MAI-1996 23.26-OCT-1995 19.51-OCT-1995

4 .194224493588E+03 .200000000000E+03 .205942056417E+03 .476282331465E+05 .469741044728E+05 .471197974987E+05

5 235 50459.782 17020.228 10.74-MAI-1996 24.91-OCT-1995 22.79-OCT-1995

5 .194241982924E+03 .200000000000E+03 .205825575194E+03 .513442055243E+05 .170208280272E+05 .233260844425E+05

1

4.5 Ranges in the Manifold Suited for the Transfer

The purpose of this section is to obtain the ranges well suited for the transfer of the parameter along the orbit on the stable manifold.

First of all the local minima selected in the previous section are improved by quadratic interpolation using the values D_{n-1} , D_n and D_{n+1} , obtained by program SELWS, as a starting point. The iterative procedure give us the displacement D_m , in the sense of the stable direction, from the point u_n in the halo orbit, which gives the local minimum distance.

Then we choose a closer distance to the center of the Earth, H_o , (in our computations we took $H_o = 7\,000$ km) and we look for the local minima which are below H_o . For them, the interval in the D parameter around D_m , (D_l^o, D_r^o) is computed in such a way that its boundary values D_l^o and D_r^o are the displacements from u_n which give the orbits in the stable manifold with the local minimum to the Earth at H_o km. In principle all the orbits between D_l^o and D_r^o could be well suited for the transfer. But due to the fact that the departure from a Geostationary Transfer Orbit is better done orthogonal to a line containing a radius of the Earth, the insertion in the orbit of the stable manifold must be done when it is just at the minimum distance. If this minimum is under a certain height, it must be discarded. Then we fix a minimum distance H_i from the center of the Earth (in our computations we took $H_i = 6\,400$ km) and we compute D_l^i and D_r^i in the same way we computed D_l^o and D_r^o . We have another interval (D_l^i, D_r^i) ¶ which must be removed from (D_l^o, D_r^o) . The obtained intervals (D_l^o, D_l^i) and (D_r^i, D_r^o) , when (D_l^i, D_r^i) is not empty, or (D_l^o, D_r^o) , when (D_l^i, D_r^i) is empty give the ranges of the manifold well suited for the transfer.

4.5.1 Main Programs and Basic Routines

This section contains two main programs:

Program AFMIN This program reads the output of the preceding two programs: SELWS and VECSWS. Then it computes the local minima of the stable manifold. The output is written in a file.

Routine RSEL This routine reads the output of program SELWS.

Routine RVECS This routine reads the output of program VECSWS.

Routine MDTWSQ It is the main routine for the computation of the local minima of the stable manifold. It uses quadratic interpolation in order to find them.

Program INTER It is the main program for the computation of the intervals well suited for the transfer orbits. It reads the output of the programs AFMIN and VECSWS and writes its output in a file.

Routine RAFIN This routine reads the output of program AFMIN.

¶ This is an empty interval if H_i is less than the minimum distance given by D_m .

Routine IWSMD It is the main routine for the computations of the intervals well suited for the transfer orbits.

4.5.2 *Output and Sample of Results*

With the results of the preceding programs, the program AFMIN produces the following output:

- Some comments related to an interactive execution of the program including some manual modification of the file (see [4]).
- The names of the files with the results of programs SELWS and VECSWS: `selws1.dat` and `vecs1.dat`.
- The parameters defining the model of solar system and the spacecraft section divided by the mass: SPCSM, IPA, ICO. For coherence, these values should be the same ones used in the previous computations.
- The zero that appears under SPCSM has the same meaning as in the preceding programs.
- The next line contains general information about the computations.
The quasi-periodic orbit corresponds to a z -amplitude $Z0 = 0.08$, and the initial epoch for its numerical refinement has been taken as $T0 = 16587$.
As in the previous output files, $NV0 = 3$ means that the revolution that has been selected for the computations is the second one.
 NTR is the number of minima that have been studied.
 MAR is an index which allows to do manual modifications to the file (see [4]).
- The following lines must be considered in groups of eight, numbered from 1 to NTR .

In the first line it appears an error flag of the routine MDTWSQ. The integer that follows this flag denotes the point of the file `estws11.dat` from which the minimum has been refined. The next three numbers are the epochs (in modified Julian days) of the departure from the qpo, closest passage to the Moon and closest passage to the Earth respectively. The last two are the minimum distances to the Moon and to the Earth that have been computed.

In the second line, the first three numbers give the last interval in the parameter along the orbit, measured on the stable direction, obtained by routine MDTWSQ. Of these three values, the second one is related to the minimum. The last three values of this line are the distances to the Earth for the values of the above parameters.

In the following three lines appear the ecliptic positions and velocities related to the three epochs given in the first line, respectively.

Finally, the last three lines have been copied from the file `vecsws1.dat`. They give the epoch, position and velocity on the qpo and stable direction on that point. All of them corresponding to the point of index given in the first line. Only a sample of results is given.

Table 4.4 Partial output of program AFMIN

OUTPUT OF PROGRAM AFMIN

If a manual change of the displacements is done, then the variable MAR must be changed according to variable IND of routine IWSMD

INPUT FILE 1 (OUTPUT OF PROGRAM SELWS1): selws1.dat

INPUT FILE 2 (OUTPUT OF PROGRAM VECWS1): vecsws1.dat

SPCSM= .1000000000E-01 IPA= 1 1 11 1 1 1 1 1 1 1 1 0 0 0 ICD= 2

0

ZO= .080000 TO= 16587.00000 NVO= 3 NTR= 5 MAR= 0

```

1 0 10 .167735940663E+05 .165669685858E+05 .165669685858E+05 .363213924388E+06 .110134765160E+05
1 .194300805609E+03 .197693266840E+03 .200000000000E+03 .110880286972E+05 .110734765160E+05 .110804394672E+05
1 .460694002504E+08 .139286653000E+09 .111571784618E+06 -.247492901751E+07 .013123560280E+06 -.101648437179E+04
1 -.970760578247E+08 -.114903978438E+09 .161266958717E+05 .259036147618E+07 -.197205922763E+07 -.200320778182E+05
1 -.970760578251E+08 -.114903978438E+09 .161266953747E+05 .259036147461E+07 -.197205923088E+07 -.200320781794E+05
1 .167735940663E+05 .000000000000E+00
1 .460693953960E+08 .139286455379E+09 .111574009889E+06 -.247492598070E+07 .813131672850E+06 -.101686717651E+04
1 .245954422931E-01 .999634114358E+00 -.11256181447E-01 -.153611962939E-01 -.410361489525E-01 .193635688203E-02
2 0 31 .167883542112E+05 .165828680526E+05 .165805312284E+05 .786056607196E+05 .144224958830E+05
2 .194461532603E+03 .198998403802E+03 .199163515573E+03 .177671630640E+05 .144224950830E+05 .144315097656E+05
2 .839492177820E+07 .146470009533E+09 .851672692504E+05 -.260051132259E+07 .154269543784E+06 -.250206365838E+04
2 -.629624171955E+08 -.136519197318E+09 .686941479186E+05 .235602274014E+07 -.990337851607E+06 .140047420116E+05
2 -.685607083246E+08 -.134156313065E+09 .167975180653E+05 .281752415401E+07 -.146858018989E+07 -.331232330642E+05
2 .167883542112E+05 .000000000000E+00
2 .839496311122E+07 .146469814890E+09 .851648323833E+05 -.260050959948E+07 .154278096084E+06 -.250229741087E+04
2 -.207705272936E+00 .978114800709E+00 .122456616110E-01 -.865893800494E-02 -.429767282348E-01 .117464507183E-02
3 0 72 .168171716369E+05 .166074743052E+05 .166044094702E+05 .619766229743E+05 .971509190949E+05
3 .194483394618E+03 .198463209061E+03 .198497196707E+03 .123088301684E+06 .971509190949E+05 .971549440183E+05
3 -.643051791444E+08 .132256114105E+09 -.113167100832E+05 -.233474804495E+07 -.111640114795E+07 -.370237090972E+04
3 -.260185969057E+07 -.150797628625E+09 .754324735858E+05 .259962824843E+07 .252837233110E+05 .157078243177E+05
3 -.107524107015E+08 -.150590310766E+09 .379958512905E+04 .2731726893329+07 -.296252050431E+06 .116880969610E+05
3 .168171716369E+05 .000000000000E+00
3 -.643050356123E+08 .132255977457E+09 -.113273670291E+05 -.233475125703E+07 -.111639810999E+07 -.370211582372E+04
3 -.723217567601E+00 .688529553366E+00 .536973375015E-01 .121537752015E-01 -.455397382082E-01 -.128530621274E-02

```

Table 4.5 Output of program INTER

```

OUTPUT OF PROGRAM INTER
INPUT FILE (OUTPUT OF PROGRAM AFMIN) : afmin1.dat
SPCSM= .100000000000E-01 IPA= 1 1 11 1 1 1 1 1 1 1 1 1 0 0 0 ICD= 2
GEOCENTRIC RADIUS OF THE INTERVAL, RACT=      7000.0000
0
ZO= .080000 TQ= 16587.00000 NVQ= 3 NTR= 1
1 0 5 235 .1693173847570E+05 .0000000000000E+00 .1955240019640E+03 .5845331363320E+04
1 -.9752097311870E+08 -.1127295578710E+09 .1454693741990E+06 .1913983251080E+07 -.1670027737650E+07 .1267670471310E+04
1 -.3171295710080E+00 -.9478202010560E+00 -.3264508634380E-01 .2331203197130E-01 .3564538264210E-01 .22624230929209-02
1 0 0 194.8895499647915 196.7478729495161 6999.996272 6999.994776

```

With the results of the preceding programs, the program INTER produces the following output:

- The name of the file with the results of program AFMIN: `afmin1.dat`.
- The parameters defining the model of solar system and the spacecraft section divided by the mass: `SPCSM`, `IPA`, `ICO`. For coherence, these values should be the same ones used in the previous computations.
- The geocentric altitude, in km, which gives the interval of the manifold that shall be considered. The orbits of the manifold which do not go below this altitude are skipped. The value taken in the example, `RACT`, is 7000 km.
- The zero that appears under this line has the same meaning as in the preceding output files.
- The next line contains general information about the computations. The quasi-periodic orbit corresponds to a z -amplitude $Z0 = 0.08$, and the initial epoch for its numerical refinement has been taken as $T0 = 16587$. As in the previous output files, $NVO = 3$ means that the revolution that has been selected for the computations is the second one. NTR is the number of intervals that have been computed.
- The following lines must be considered in groups of four, numbered from 1 to NTR .

In the first line appears a general error flag of the routine `IWSMD`. The integer that follows this flag denotes the point in the file `afmin1.dat` from which the interval has been computed. The next integer in the line denotes the point of the file `estws11.dat` from which the minimum in file `afmin1.dat` has been refined. The next two values together with the second and the third lines are the same ones that appeared in the last three lines, of each group, of the file `afmin1.dat` from which they have been copied.

The remaining two values of the first line are the value of the parameter along the orbit in the stable direction related to the minimum distance to the Earth and this minimum distance in km.

The fourth line has two error flags, the values of the parameter along the orbit which define the interval and the distances to the Earth associated to these values of the parameter (these values must be roughly `RACT`).

4.6 Characteristics of the Orbits Near the Earth

Finally, the last step is to study the characteristics of the orbits contained in the intervals computed in the previous section and to plot the results.

The orbital elements of the orbit, with respect to the Earth, are computed because they give the necessary and sufficient information about the orbit in the neighborhood of the Earth. Moreover they hardly change near the Earth. Other things such as periapsis height, epochs of insertion, and distance from the orbit to the Moon are also computed.

A thing to take into account is that in our computations the Earth has been considered as if its mass were concentrated at a single point. A good thing would be to use a more realistic Earth's potential when the orbit comes near to the Earth.

4.6.1 Main Program and Basic Routines

The program and routines involved in this step are the following:

Program FINES It is the main program for the computations of the characteristics of the orbits. It reads the output of program INTER and writes its output in a file.

Routines RINTPA and RINTER They are used for reading the output of program INTER.

Routine HCTWS This routine takes an orbit in the interval obtained by program INTER, and follows it backwards in time till it is at a certain fixed height from the center of the Earth.

Routine PVELEM This routine is used to compute the orbital elements from the position and velocity given by routine HCTWS.

4.6.2 Output and Sample of Results

With the results of the preceding programs, the program FINES produces the following output:

- The names of the files with the output of program INTER : `inter11.dat` and `inter21.dat`.
- The parameters defining the model of solar system and the spacecraft section divided by the mass: `SPCSM`, `IPA`, `IC0`. For coherence, these values should be the same ones used in the previous computations.
- The geocentric altitudes, in km, as defined in program INTER. In the example the values taken are `RACT1 = 7000` and `RACT2 = 6400`.
- The geocentric altitude at which the Keplerian elements are computed (`RCE = 7000 km`).
- The zero that appears under this line has the same meaning as in the

preceding output files.

- The next line contains general information about the computations. The quasi-periodic orbit corresponds to a z -amplitude $Z0 = 0.08$, and the initial epoch for its numerical refinement has been taken as $T0 = 16587$. As in the previous output files, $NVO = 3$ means that the revolution that has been selected for the computations is the second one. NTR is the number of intervals that have been considered.

- The next line contains: $IPUN$ which is the pointer to the initial point for the computations of the interval in file `estws11.dat`, TW the epoch of departure from the qpo and NSI which is the number of subintervals mentioned in the previous section.

- According to the value of NSI , there appear one or two large groups of results. Each one contains in the first line the interval of the parameter along the orbit ($XLAI$, $XLAF$) and the number of points that have been taken in this interval (JBN). For each one of these points a group of 6 lines is given with the following data:

In the first line an index ranging from 1 to NTR and the index JBN . The epoch of passage by the periapsis, the epoch of passage at a distance of 7 000 km (RCE) and the epoch of passage at minimum distance from the Moon (all of them in MJD). The geocentric altitude of the periapsis and the geocentric altitude where the orbital elements have been computed (both in km). The tangent impulse (from a GTO circular parking orbit) for the departure (in m/s). The velocity at the periapsis and the velocity at the geocentric distance RCE , both in km/s and with respect to the Earth. The minimum distance from the transfer orbit to the Moon (km) and the angle between the Earth-Sun line and Earth-Moon line (in degrees) at the departure. The sign of these angles is positive if the Moon is in the first quarter and negative if in the third.

The second line contains the Keplerian elements at the moment of geocentric distance RCE : The semilatus rectum (km), the eccentricity, the inclination, the argument of the ascending node, the argument of perigee, and the true anomaly (all the angles in degrees).

The third line gives the position (km) and velocity (km/day) at the periapsis with respect to the center of the Earth.

The fourth line gives the position and velocity of the Earth at the epoch at which the satellite is in the periapsis (in barycentric ecliptic coordinates and in km, km/day).

The fifth line gives the position and velocity of the satellite at the epoch of minimum distance to the Moon (with the same units as above).

And the last line gives the position and velocity of the Moon at the same epoch and with the same coordinates as in the fifth line.

OUTPUT OF PROGRAM FINES

INPUT FILE (OUTPUT OF PROGRAM INTER): inter11.dat

INPUT FILE (OUTPUT OF PROGRAM INTER): inter21.dat

SPCSM= .10000000000E-01 IPA= 1 1 11 1 1 1 1 1 1 1 1 0 0 0 ICQ= 2

GEOCENTRIC RADIUS OF THE INTERVAL IN THE INPUT FILES, RACT1= 7000.0000 km. and RACT2= 6400.0000 km.

ORBITAL ELEMENTS COMPUTED AT GEOCENTRIC DISTANCE, RCE= 7000.0000 km.

0

ZO= .080000 TO= 16587.00000 NVO= 3 NTR= 1

IPUN=235 TW= 16931.7385 NSI= 2

INTERVAL IN LAMBDA: XLAI= 194.8895500 XLAF= 194.9217593 JBN= 20

1	0	16130.0017653	16730.0017661	16732.5665	6999.9963	6999.996	3054.46	10.60051	10.60051	9948.6019	28.34
		13813.8	.97339	80.20	34.57	8.94	.01				
		.558911658859E+04	.407601026637E+04	.10713e409589E+04	-.204531982464E+06	.461104426572E+05	.891563037096E+06				
		.131660783439E+09	.698316535033E+08	.727821339051E+03	-.123082253060E+07	.227520904487E+07	-.291438352299E+03				
		.128073373411E+09	.753944104902E+08	.487129027762E+03	-.142097318343E+07	.215741087848E+07	-.895809063720E+04				
		.128072129875E+09	.753858438381E+08	.539026786653E+04	-.127657219565E+07	.214620209913E+07	.808147113196E+04				
1	1	16730.0027785	16730.0038366	16732.5666	6966.6947	7000.002	3062.11	10.62610	10.60053	9961.0665	28.32
		13749.0	.97353	80.13	34.57	8.92	7.96				
		.556231904472E+04	.405751150330E+04	.106397467334E+04	-.205439755237E+06	.473161602931E+05	.893569764619E+06				
		.131659536417E+09	.698339586074E+08	.727526065697E+03	-.123086294993E+07	.227518716067E+07	-.291446588132E+03				
		.128073247281E+09	.753946340759E+08	.485698213232E+03	-.142095210773E+07	.215738495575E+07	-.892560915782E+04				
		.128072004831E+09	.753060540645E+08	.539105946772E+04	-.127657483931E+07	.214620112673E+07	.808143960039E+04				
1	2	16730.0037920	16730.0052825	16732.5667	6933.4800	7000.000	3069.80	10.65196	10.60055	9973.5767	28.31
		13684.3	.97366	90.05	34.57	8.90	11.26				
		.553558937076E+04	.403905886579E+04	.105659769173E+04	-.206357041041E+06	.485354719349E+05	.895582322539E+06				
		.131658208912E+09	.698362645081E+08	.727230679243E+03	-.123090338325E+07	.227516626929E+07	-.291454822003E+03				
		.128073120790E+09	.753948583223E+08	.484265256213E+03	-.142093099955E+07	.215735899103E+07	-.889312657430E+04				
		.128071879422E+09	.753062649054E+08	.539185337959E+04	-.127657689901E+07	.214620015253E+07	.808140791175E+04				
1	3	16730.0047978	16730.0066139	16732.5668	6900.6150	7000.000	3077.46	10.67766	10.60057	9986.0330	28.30

Table 4.6 Partial output of program FINES

4.6.3 Some Numerical Explorations

Several runs have been done with the above mentioned software. We present here the results obtained, associated to the quasi-periodic halo orbits of z -amplitude 0.08, for values of t_0 , the initial epoch used for the refinement of the qpo, between the epochs 16 470 and 16 705 (MJD).

The following table contains some insertion (on the stable manifold) windows found for this time interval:

T_i	T_f	ΔT
16462.45474	16464.33193	1.87719
16491.95449	16493.79283	1.83834
16522.09843	16523.24531	1.14688
16608.41647	16610.88113	2.46466
16669.11615	16669.81857	0.70242
16698.87606	16700.47233	1.59627

Table 4.7 Some insertion windows between 16 462 and 16 700 (MJD). T_i is the opening date of the window, T_f is the closing date and ΔT is the width of the window.

Figures 4.2 to 4.8 show several windows with data related to the transfer orbit. They give information about several magnitudes such as epoch, inclination, velocity with respect to the Earth, Δv for the injection from a circular orbit, minimum distance to the Moon along the transfer orbit and Sun–Earth–Moon angle at the insertion epoch for a range of perigee between 6 400 and 7 000 km. We remark that the insertion windows have been selected looking only at perigee distances. In fact one should also add the requirements concerning the admissible inclinations. The distances to the Moon should be as large as possible within the window to prevent from too strong manoeuvres to correct the errors of the injection manoeuvre.

Figures 4.9 to 4.13 display those magnitudes against the MJD-16462 for a value of the perigee equal to 6 564.1 km.

Figure 4.14 represents a transfer orbit to the halo orbit with $t_0 = 16470$ and $\beta = 0.08$. The perigee distance for that transfer orbit is roughly 6 870 km and the ecliptic inclination is roughly 24° . The transfer takes 208.1 days from the injection till a distance of 197.5 km to the halo orbit, but it takes less than 170 days to arrive at 1 000 km from the halo.

Finally Figure 4.15 shows the continuation of the transfer orbit given in the Figure 4.14 for 7 additional months.

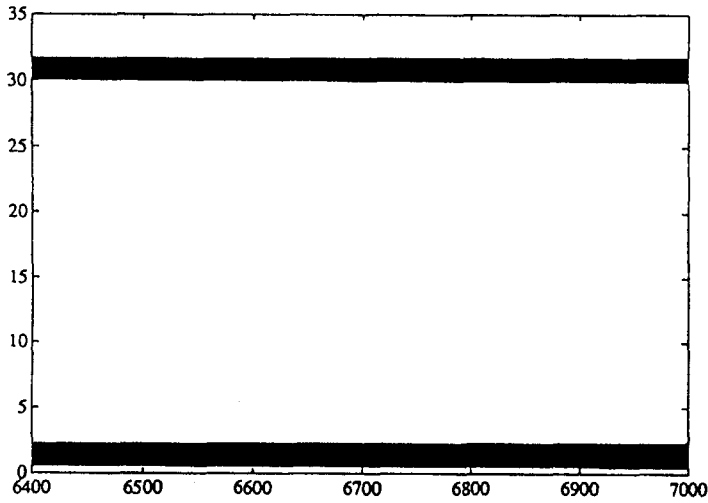


Fig. 4.2 Insertion windows. The abscissae are the perigee distances in km. The ordinates are the MJD minus 16462. For each perigee distance the admissible insertion window consists of several intervals. Here two of them are shown. Most of the other intervals are spaced by roughly 30 days.

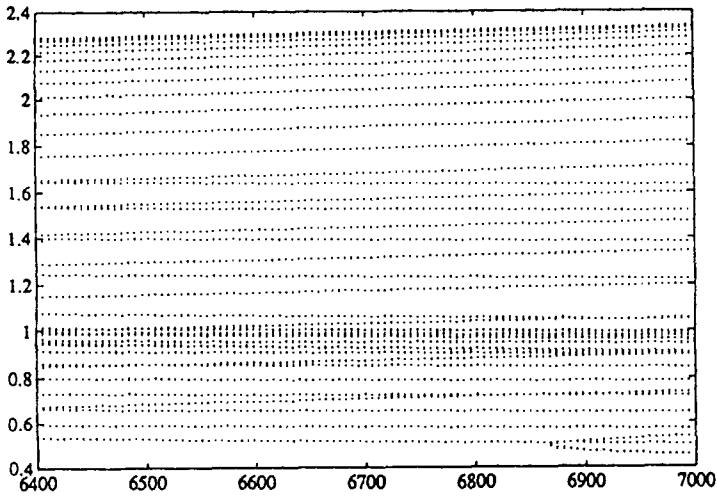


Fig. 4.3 Same as Figure 4.2 but restricted to the window displayed at the bottom of Figure 4.2. In fact the strip between the lower and the upper (dotted) curves displayed here is full of curves. For convenience we only display some of them. Each, roughly horizontal, dotted curve corresponds to a given value of the initial value, t_0 , of the refined halo orbit. Along each one of those curves the points are obtained by changing the parameter along the orbit. We remark that given t_0 the corresponding curve cuts twice the perigee range 6400 to 7000 km. For instance, on the right lower part of the figure one can see the minimum (of the perigee distance) of one of those curves. The minima of the other ones are located to the left of the figure, outside the range of interest.

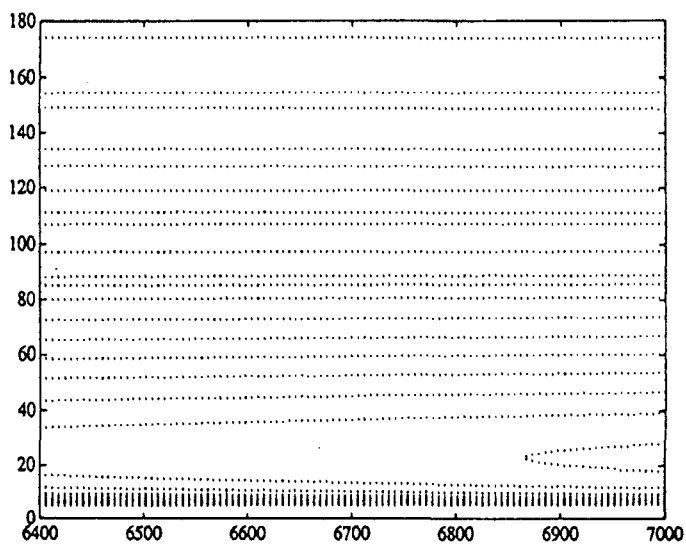


Fig. 4.4 Same as Figure 4.3 displaying the inclination in degrees with respect to the ecliptic plane instead of the MJD.

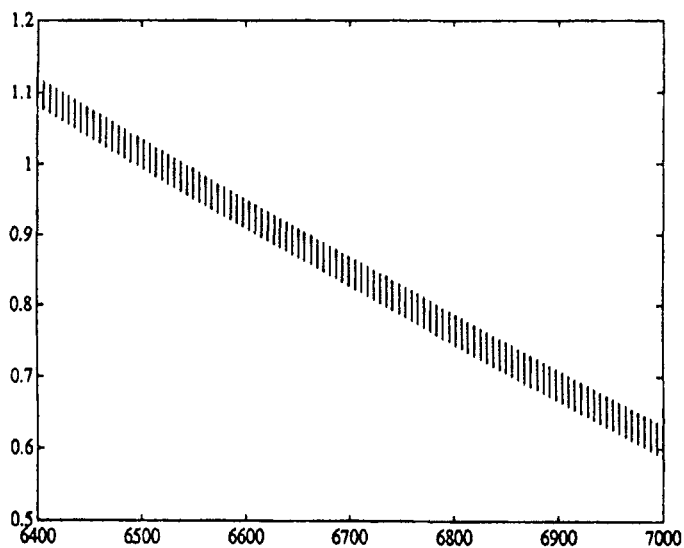


Fig. 4.5 Same as Figure 4.3 displaying in ordinates the velocity with respect to the Earth at the perigee in km/s after subtracting 10 km/s.

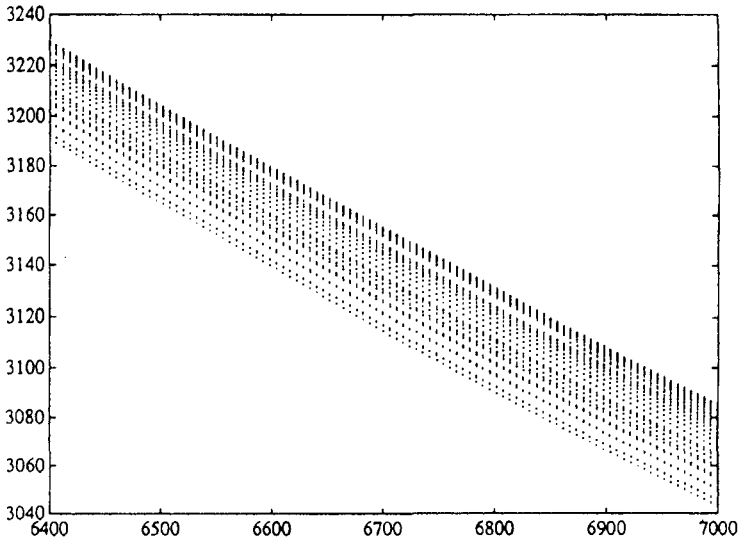


Fig. 4.6 Same as Figure 4.3 displaying in ordinates the Δv required at the perigee to insert into the transfer orbit from a circular orbit. It is given in m/s.

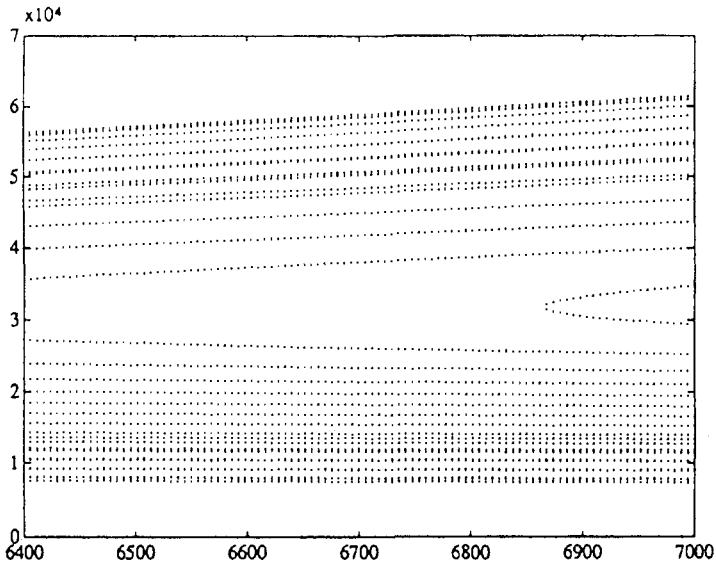


Fig. 4.7 Same as Figure 4.3 displaying in ordinates the minimum distance to the Moon, in km, along the transfer orbit.

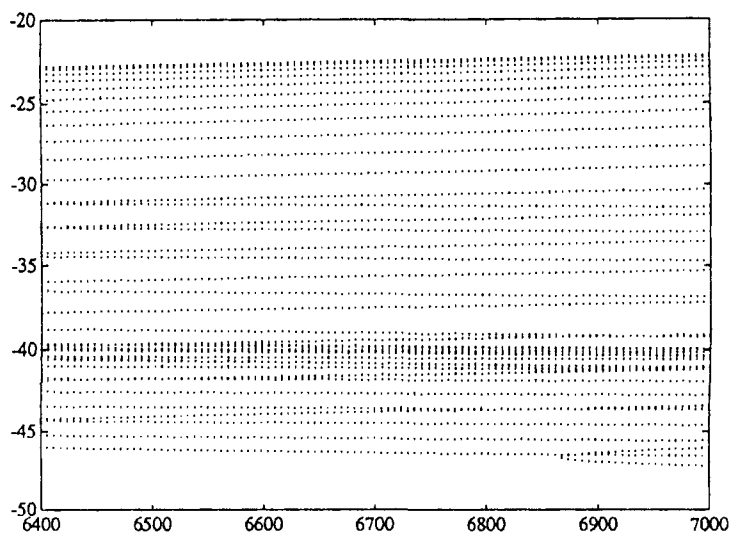


Fig. 4.8 Same as Figure 4.3 displaying in ordinates the Sun-Earth-Moon angle, in degrees, at the moment of the insertion into the transfer orbit. The negative angle means that the Moon is waning.

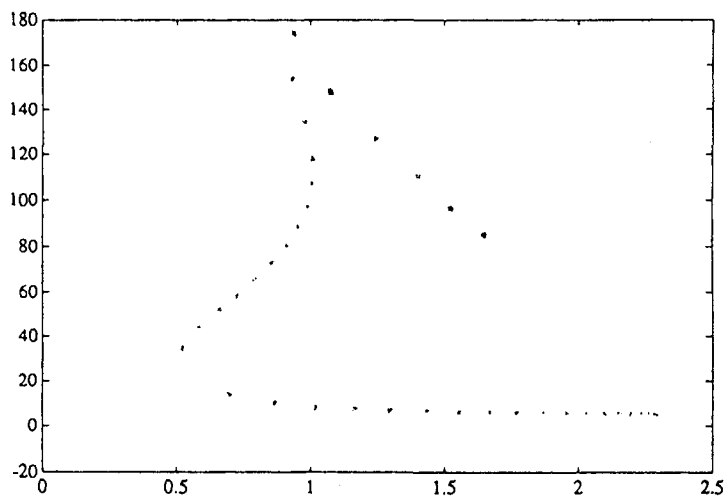


Fig. 4.9 For the selected perigee distance 6 564.1 km (which was the value taken for the ISEE-3 mission) we display in abscissae the MJD 16 462, and in ordinates the ecliptic inclination.

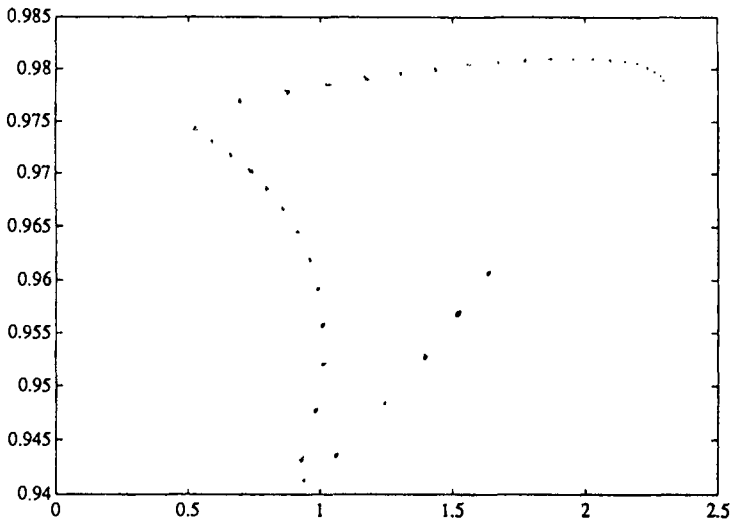


Fig. 4.10 Same as Figure 4.9 but in ordinates the relative velocity to the Earth is shown (in km/s minus 10 km/s).

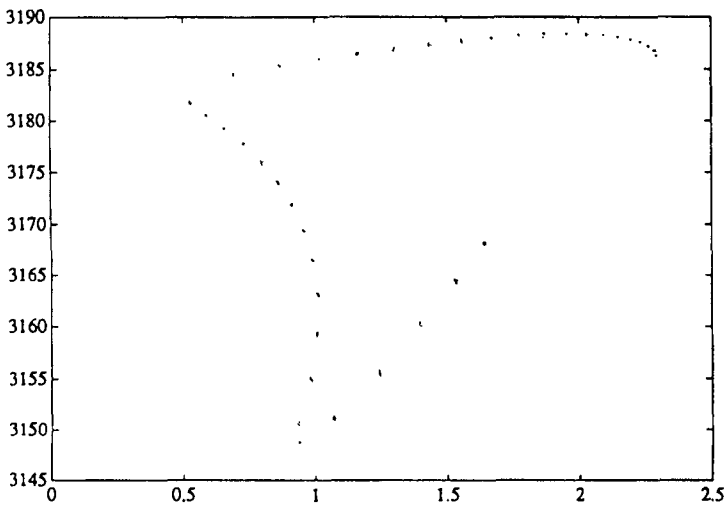


Fig. 4.11 Same as Figure 4.9 but in ordinates the required Δv to inject into the transfer orbit from a circular one is given (in m/s).

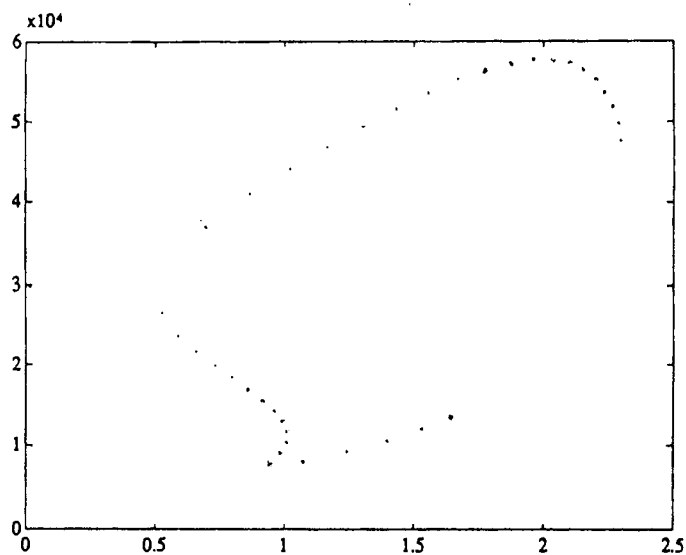


Fig. 4.12 Same as Figure 4.9 but in ordinates the minimum distance to the Moon (in km) along the transfer orbit is shown.

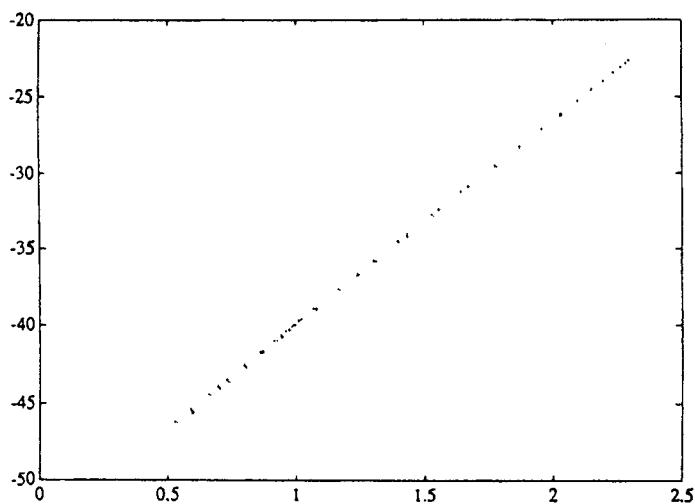


Fig. 4.13 Same as Figure 4.9 but in ordinates the Sun-Earth-Moon angle at the insertion epoch is displayed.

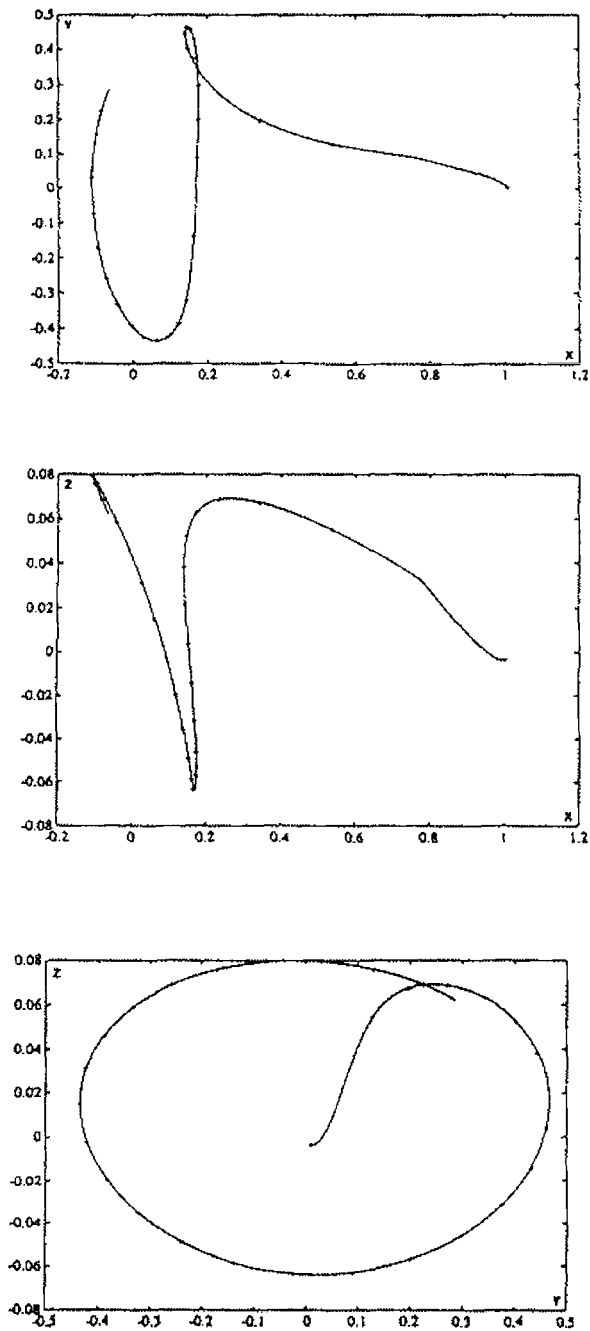


Fig. 4.14 Transfer orbit for a halo orbit with $t_0 = 16\,470$ MJD and $\beta = 0.08$. Epoch of insertion into the transfer orbit: 16 462.4603 MJD. Arrival to 197.5 km of the halo orbit along the stable direction at the epoch 16 670.5603 MJD. The crosses along the orbit denote intervals of 7 days starting at the insertion epoch.

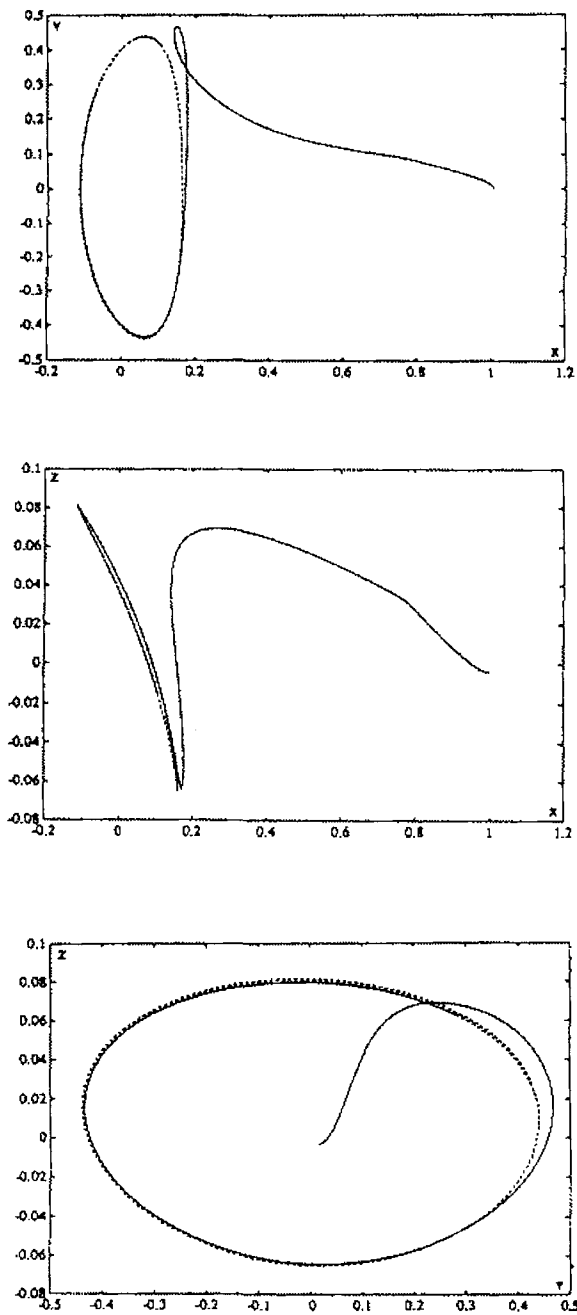


Fig. 4.15 Same as Figure 4.14 but continuing the integration for more than 200 days.

4.7 Conclusions

The transfer from the vicinity of the Earth to a halo orbit near the L_1 point of the Earth+Moon-Sun system is a well-known topic. However the approach using routinely a general optimization procedure gives no insight on the reasons explaining the nature of the orbit.

The Dynamical Systems approach, using the stable manifold of the orbit, gives a way to produce possible transfer orbits when the target orbit is unstable. In other words, we are asking the proper dynamics of the motion to help us in the transfer process.

Appendix A

The JPL Model

A.1 Implementation and Use

A direct access file with the JPL DE-118 ephemeris has been implemented in our computer (HP-9000/835S). The file contains 709 records of 5960 bytes, and each record has data of the bodies of the solar system for 32 days. The dates for which it can be used range between 1959/12/10 = (MJD1950) 3 630 and 2022/01/21 = (MJD1950) 26 318.

As it is known, what is stored in the file are not the coordinates of the planets (day by day, for instance) but the coefficients to be multiplied by the Chebyshev polynomials for different time intervals (typically of 32 days except for Mercury (8 days), the Earth (16 days) and the Moon (4 days)) to obtain a representation of the desired quantities.

Making use of the recursive relations for the evaluation of the Chebyshev polynomials, as well as their derivatives, the positions, velocities, accelerations and overaccelerations of the bodies can be easily computed. All the planetary coordinates are given in the mean equatorial system of 1950.0 with origin at the solar system mass center, except the Moon which is geocentric.

A routine was provided to us by the Agency (routine PLANET) which, from the JPL ephemeris, gives the position (km) and velocity (km/s) of the bodies of the solar system in the above mentioned reference system. This routine has been modified in two points:

- (1) It computes also the acceleration and overacceleration of a given body. This is necessary because in the computation of halo orbits several reference systems are used: ecliptic, adimensional (natural extension of the synodic system of the Restricted Three Body Problem) and normalized (well suited for the computation of orbits around the libration points). The transformation of coordinates between the first two reference systems, involves the velocity of one primary with respect to the other (i.e. the Earth–Moon barycenter with respect to the Sun), so when velocities must be transformed from eclip-

tic to adimensional coordinates, or vice versa, the accelerations are needed (see [2] pp 354-358). Similarly, when the accelerations must be transformed the overaccelerations are needed.

- (2) The second modification is a minor one related to the units used, which are: km, km/day, $\text{km}/(\text{day})^2$, $\text{km}/(\text{day})^3$ for the position, velocity, acceleration and overacceleration respectively.

The adopted constants for the computation of trajectories are the same as the ones of the JPL ephemeris, and are the following:

Astronomical Unit : 149 597 870.66 km.

Earth/Moon mass ratio : 81.300587.

Solar radiation pressure coefficient : $4.51 \times 10^{-9} \text{ kg km} / (\text{m s})^2$ (this is the force acting per unit of surface at one AU).

Gravity potentials of the bodies in $\text{AU}^3 / \text{day}^2$:

Mercury	0.49125474514508e-10
Venus	0.72434562096328e-9
Earth+Moon	0.89970116585573e-9
Mars	0.95495289422241e-10
Jupiter	0.28253421034459e-6
Saturn	0.84594685048307e-7
Uranus	0.12888162381380e-7
Neptune	0.15321124812843e-7
Pluto	0.22762477518637e-11
Moon	(Earth+Moon)/82.300587
Sun	0.29591220828559e-3

Several tests have been done between the numerical ephemeris and our previously developed analytic model of the solar system (program SS, see [2], V.1). The following table summarizes the results obtained for the differences (in degrees) between the analytically computed longitude and latitude of the bodies of the solar system (with respect to the barycenter of the full solar system) and the values obtained from the numerical ephemeris.

Differences in longitude

Body	Max. Diff.	Average	Variance
Mercury	-0.3326e-1	0.2013e-2	0.2915e-3
Venus	0.1937e-1	0.4112e-3	0.8394e-4
Earth	-0.1162e-1	-0.1862e-2	0.4098e-4
Mars	0.3245e-1	-0.1123e-2	0.1009e-3
Jupiter	0.1472	0.2295e-1	0.4472e-2
Saturn	0.2101	0.6477e-1	0.1134e-1
Uranus	-0.9455	-0.8498	0.3259e-2
Neptune	0.5862	0.5659	0.2032e-3
Moon	-0.1163e-1	-0.1862e-2	0.4102e-4
E-M Barycenter	-0.1162e-1	-0.1862e-2	0.4098e-4

Differences in latitude

Body	Max. Diff.	Average	Variance
Mercury	-0.1082e-1	-0.6733e-2	0.1589e-5
Venus	-0.4181e-2	-0.3490e-2	0.8955e-7
Earth	-0.2856e-2	-0.2704e-2	0.6843e-8
Mars	-0.2349e-2	-0.1749e-2	0.6362e-7
Jupiter	-0.3165e-2	-0.3700e-3	0.1619e-5
Saturn	0.7271e-2	0.1017e-2	0.1340e-4
Uranus	0.1017e-1	0.5473e-2	0.1307e-4
Neptune	-0.1771e-1	-0.1692e-1	0.1127e-5
Moon	-0.2892e-2	-0.2704e-2	0.7209e-8
E-M Barycenter	-0.2856e-2	-0.2704e-2	0.6842e-8

The time interval that has been used is $1990/01/01 = (\text{MJD}1950) 14610$, $2010/01/01 = (\text{MJD}1950) 21\,016$ and the ephemeris have been computed day by day. The comparison between the positions, velocities, accelerations and overaccelerations gives results that are also in agreement, but some differences appear because the values adopted for the different main constants, such as the Astronomical Unit and the masses of the bodies are slightly different in both programs.

It must also be said that the ephemeris obtained by our analytic model are given in the mean ecliptic reference system of the epoch at which they are computed (the effect of the variation of the equinoxes was not taken into account in program SS). So some transformations must be done in order to reduce them to the mean equatorial system of 1950.0, which is the one used by the JPL ephemeris, as it has been said. The values of the parameters used for the reduction are those of

the “Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac”:

- For the obliquity

$$\epsilon = 23^{\circ}27'08''.26 - 46''.845T - 0''.0059T^2 + 0''.00181T^3,$$

where T is measured in tropical centuries from 1900.0 .

- For the precession

$$\zeta_0 = (2304''.250 + 1''.396T_0)T + 0''.302T^2 + 0''.018T^3,$$

$$z = \zeta_0 + 0''.791T^2,$$

$$\theta = (2004''.682 - 0''.853T_0)T - 0''.426T^2 - 0''.042T^3,$$

where T_0 is the initial epoch and $T_0 + T$ the final one, both in tropical centuries from 1900.0 .

If \vec{x}_0 denotes the position vector with respect to the epoch T_0 and \vec{x} the corresponding one with respect to the epoch $T_0 + T$, they are related by

$$\vec{x} = P\vec{x}_0,$$

where the components of the orthogonal precession matrix P are

$$P_{11} = \cos \zeta_0 \cos \theta \cos z - \sin \zeta_0 \sin z,$$

$$P_{12} = -\sin \zeta_0 \cos \theta \cos z - \cos \zeta_0 \sin z,$$

$$P_{13} = -\sin \theta \cos z,$$

$$P_{21} = \cos \zeta_0 \cos \theta \sin z + \sin \zeta_0 \cos z,$$

$$P_{22} = -\sin \zeta_0 \cos \theta \sin z + \cos \zeta_0 \cos z,$$

$$P_{23} = -\sin \theta \sin z,$$

$$P_{31} = \cos \zeta_0 \sin \theta,$$

$$P_{32} = -\sin \zeta_0 \sin \theta,$$

$$P_{33} = \cos \theta.$$

A.2 Basic Subroutines

Subroutine COSS This routine stores the coordinates, in the mean ecliptic system of 1950.0, of the planets and the Moon at a given epoch with respect to the Sun and with respect to the barycenter of the solar system. The storing of the above mentioned ephemeris is done in an identical way as it is done when the analytic model of solar system is used.

Subroutine EPHJPL This routine uses routines `plajpl` and `equecli` in order to get the coordinates of the bodies of the solar system in the suitable way used by routine `cooss`. In fact this routine is a bridge between `plajpl` and `cooss` in order to use `cooss` in the same way as it is used in [2].

Subroutine EQUCLI This routine performs the transformation from mean equatorial coordinates at 1950.0 to the mean ecliptic reference system also of 1950.0 . For the computations, the values of the parameters used are the

ones of the “ Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac ”.

Subroutine JULCAL This routine transforms modified Julian dates to calendar dates (day/month/year) or vice versa.

Subroutine MODJPL This routine initializes all the basic constants such as the mass ratios of the planets, conversion factors for distances, angles, etc. It is used to select the bodies of the solar system in order to define the model to be used during the computations. Its role is similar to the one performed by routines `model1` and `model2` (see [2]) used when an analytic model of motion for the bodies of the solar system is used. The values of the basic constants are the ones of the JPL tapes and are slightly different from the ones used in the analytic model (see [2]).

Subroutine PLAJPL This routine is a modification of routine `planet` provided to us by the Agency. It reads the JPL ephemeris from a given file and computes the positions, velocities, ... of the bodies of the solar system at a given epoch. All the planetary coordinates are computed in the mean equatorial system of 1950.0 with origin at the solar system mass center, except the Moon which is geocentric.

This page is intentionally left blank

Appendix B

Reference Systems and Equations of Motion

B.1 The Restricted Three Body Problem

We consider in the space x, y, z , two primaries with masses m_1 and m_2 respectively. We assume that, under the influence of their mutual gravitational attraction, they revolve on a circular orbit in the (x, y) -plane around their center of masses with angular velocity equal to n (see Figure B.1).

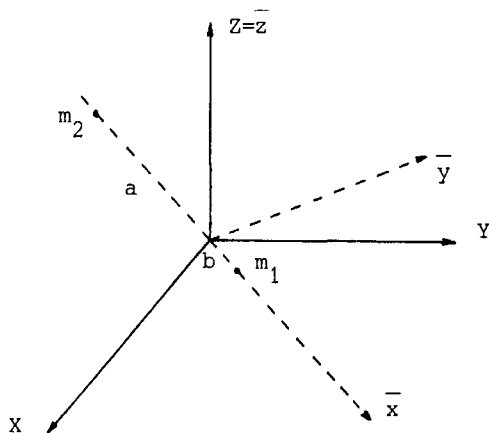


Fig. B.1 Sidereal and synodic reference frames for the restricted three-body problem.

A third body of mass m_3 much smaller than either m_1 or m_2 , moves in the space under the influence of the two primaries. In fact we consider that m_3 does not act on m_1 and m_2 .

To study the motion of m_3 the equations can be written in a suitable rotating (synodic) coordinate system $\bar{x}, \bar{y}, \bar{z}$ such that the (\bar{x}, \bar{y}) -plane rotates with the two primaries in order to have m_1 and m_2 fixed on the \bar{x} -axis. In the computations we

use the equations referred to a synodic system x, y, z of dimensionless coordinates defined by

$$x = \frac{\bar{x}}{d}, \quad y = \frac{\bar{y}}{d}, \quad z = \frac{\bar{z}}{d},$$

where $d = a + b$ (see Figure B.1) is the distance between the primaries. In this system m_1 and m_2 are located at $P_1 = (\mu, 0, 0)$ and $P_2 = (\mu - 1, 0, 0)$ respectively, where $\mu = m_2/M$ and $M = m_1 + m_2$. From now on we consider $\mu \leq 1/2$. For the applications μ is rather small. Furthermore the unit of time is taken in such a way that $n = 1$.

In this rotating, barycentric, dimensionless coordinate system, the differential equations of motion for the circular restricted three-body problem take the form

$$\ddot{\vec{x}} = \vec{f}(x),$$

where

$$\begin{aligned} \vec{x} &= (x_1, x_2, x_3, x_4, x_5, x_6)^T \\ &= (x, y, z, \dot{x}, \dot{y}, \dot{z})^T, \\ \vec{f} &= (f_1, f_2, f_3, f_4, f_5, f_6)^T \\ &= (x_4, x_5, x_6, 2x_5 + \frac{\partial \Omega}{\partial x_1}, -2x_4 + \frac{\partial \Omega}{\partial x_2}, \frac{\partial \Omega}{\partial x_3})^T \end{aligned}$$

and

$$\begin{aligned} \Omega &= \frac{1}{2}(x_1^2 + x_2^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2}, \\ r_1^2 &= (x_1 - \mu)^2 + x_2^2 + x_3^2, \\ r_2^2 &= (x_1 - \mu + 1)^2 + x_2^2 + x_3^2. \end{aligned}$$

These equations have a first integral, which is called the Jacobi integral, given by

$$C = 2\Omega - (x_4^2 + x_5^2 + x_6^2),$$

where C is the so called Jacobi constant.

It is well-known that the above system of differential equations has five equilibrium points, that are the real roots of the simultaneous equations

$$f_i = 0, \quad i = 1 \dots 6.$$

Setting $x_2 (= y) = 0$ we can evaluate the position of the collinear Lagrangian points L_1, L_2, L_3 . Here we have designated the point L_1 to lie between the primaries while L_2 rests to the left of the smaller primary and L_3 to the right of the other one. The remaining equilibrium points L_4 and L_5 form an equilateral triangle with the two primaries as it is shown in B.2.

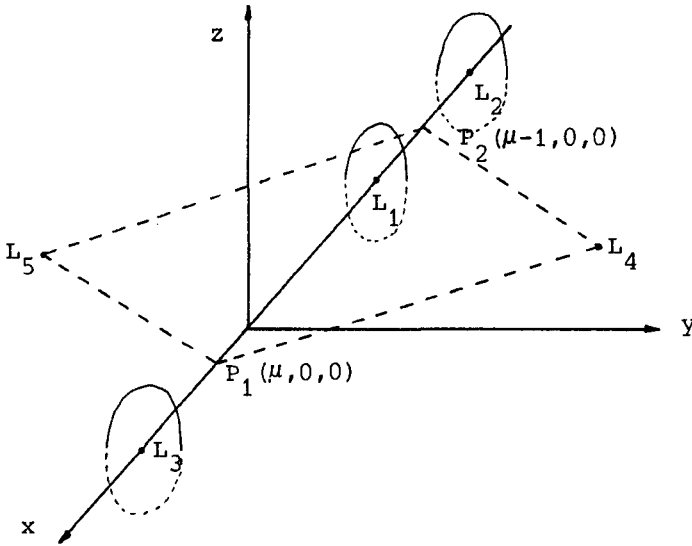


Fig. B.2 The five equilibrium points of the restricted three-body problem.

The collinear equilibrium points are unstable and each one has a family of planar hyperbolic periodic orbits (the Lyapunov families) associated to it.

A halo orbit is a spatial periodic orbit of the RTBP in synodic coordinates x, y, z such that its projection on the (y, z) -plane is homeomorphic to a circumference with the origin contained in the bounded component determined by the circumference (see Figure B.2). This kind of orbits can be obtained from the Lyapunov families associated to the equilibrium points when the frequencies on the (x, y) -plane and the vertical one are identical. In other cases the projection of the orbit on the (y, z) -plane describes a Lissajous curve (for most of the initial conditions close enough to the collinear point). The halo orbits are symmetrical respect to the (x, z) -plane. Besides one of these orbits there exist the symmetrical one with respect to the $z = 0$ plane.

In fact, for each collinear point there is one family of halo orbits emanating from a planar periodic orbit belonging to the Lyapunov family, the so called Lyapunov orbit with critical vertical stability.

For the study of halo orbits Richardson [7], gives the following Lagrangian formulation. In a reference system, centered at any of the three collinear equilibria ($E = L_{1,2,3}$), and with the x -axis, in all the cases, oriented like \vec{r}_{12} as in Figure B.3, the Lagrangian of the RTBP can be written in the form

$$L = \frac{1}{2}(\dot{\vec{\rho}}, \dot{\vec{\rho}}) + \frac{(\vec{r}_1, \vec{\rho})}{r_{12}^3} - (\dot{\vec{r}}_1, \dot{\vec{\rho}}) + \mu \left(\frac{1}{|\vec{r}_1 - \vec{\rho}|} - \frac{(\vec{r}_1, \vec{\rho})}{r_1^3} \right)$$

$$\begin{aligned}
& + (1 - \mu) \left(\frac{1}{|\vec{r}_2 - \vec{\rho}|} - \frac{(\vec{r}_2, \vec{\rho})}{r_2^3} \right) \\
= & \frac{1}{2} (\ddot{\vec{\rho}}, \ddot{\vec{\rho}}) - \frac{d}{dt} (\dot{\vec{r}}_1, \dot{\vec{\rho}}) + \mu \left(\frac{1}{|\vec{r}_1 - \vec{\rho}|} - \frac{(\vec{r}_1, \vec{\rho})}{r_1^3} \right) \\
& + (1 - \mu) \left(\frac{1}{|\vec{r}_2 - \vec{\rho}|} - \frac{(\vec{r}_2, \vec{\rho})}{r_2^3} \right),
\end{aligned}$$

where \vec{r}_1 , \vec{r}_2 , and $\vec{\rho}$ are now the positions of the primaries and the infinitesimal body, respectively, with respect to the collinear equilibrium, $\vec{r}_{12} = \vec{r}_2 - \vec{r}_1$ and the norm of a vector \vec{v} is denoted by v or by $|\vec{v}|$.

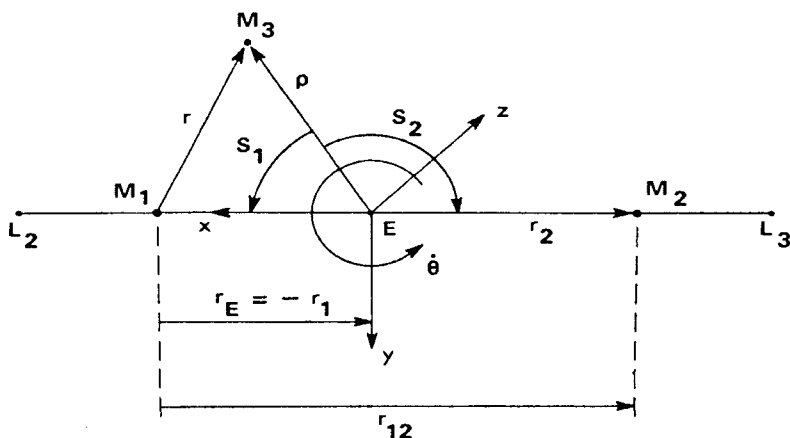


Fig. B.3 Richardson's reference system centered at the equilibrium point L_1 .

The terms involving the factors μ and $(1 - \mu)$ have the form of a third body perturbing potential. Using power series developments in terms of Legendre polynomials, the Lagrangian can be rewritten as

$$\begin{aligned}
L = & \frac{1}{2} (\ddot{\vec{\rho}}, \ddot{\vec{\rho}}) - \frac{d}{dt} (\dot{\vec{r}}_1, \dot{\vec{\rho}}) + \frac{\mu}{r_1} \sum_{n=2}^{\infty} \left(\frac{\rho}{r_1} \right)^n P_n(\cos S_1) \\
& + \frac{1 - \mu}{r_2} \sum_{n=2}^{\infty} \left(\frac{\rho}{r_2} \right)^n P_n(\cos S_2),
\end{aligned}$$

where we refer to the Figure B.3 for the meaning of the angles S_1 and S_2 .

Taking the unit of distance as:

- $r_1 = 1$ (for the motion around L_1 or L_2),
- $r_2 = 1$ (for the motion around L_3),

the Lagrangian can be written in the form:

$$L = \frac{1}{2}(\dot{\vec{\rho}}, \dot{\vec{\rho}}) + \sum_{n=2}^{\infty} c_n \rho^n P_n \left(\frac{x}{\rho} \right),$$

where primes denote derivatives with respect to the variable $s = \gamma_L^{3/2} t$, γ_L being r_1/r_{12} for L_1, L_2 and r_2/r_{12} for L_3 . We keep $\vec{\rho}$ to denote the distance of the infinitesimal mass to the collinear equilibrium after the change of scale and x, y, z are now the components of $\vec{\rho}$.

The constants c_n are given by the expressions:

- For L_1 and L_2

$$c_n = \frac{1}{\gamma_L^3} \left((\pm 1)^n \mu + \frac{(-1)^n (1 - \mu) \gamma_L^{n+1}}{(1 \mp \gamma_L)^{n+1}} \right),$$

with the upper sign for L_1 and the lower sign for L_2 ,

- For L_3

$$c_n = \frac{1}{\gamma_L^3} \left(1 - \mu + \frac{\mu \gamma_L^{n+1}}{(1 - \gamma_L)^{n+1}} \right),$$

and the equations of motion have the following compact expressions:

$$\begin{aligned} x'' - 2y' - (1 + 2c_2)x &= \sum_{n=2}^{\infty} (n+1) c_{n+1} \rho^n P_n \left(\frac{x}{\rho} \right), \\ y'' + 2x' + (c_2 - 1)y &= \sum_{n=2}^{\infty} c_n y \rho^{n-2} \tilde{P}_n \left(\frac{x}{\rho} \right), \\ z'' + c_2 z &= \sum_{n=2}^{\infty} c_n z \rho^{n-2} \tilde{P}_n \left(\frac{x}{\rho} \right), \end{aligned}$$

where

$$P_n \left(\frac{x}{\rho} \right) = \sum_{k=0}^{\lfloor \frac{n-2}{2} \rfloor} P_{n-2k-2} \left(\frac{x}{\rho} \right).$$

B.2 The Real Problem

B.2.1 Reference Systems

When dealing with the real problem several reference systems are used. The first is an ecliptic one with origin at the instantaneous center of masses of the solar system. The second one is a natural extension of the coordinate system used in the RTBP. We call it the adimensional system.

We denote by \vec{e} and \vec{a} the coordinates of a point in the ecliptic and adimensional systems of reference, respectively. We go from one system to the other one via a transformation of the form

$$\vec{e} = kC\vec{a} + \vec{b},$$

where k (change of scale factor), C (orthogonal matrix) and \vec{b} (translation) must be computed.

When we have selected two primaries, for example the Earth-Moon barycenter and the Sun, \vec{b} is the position of the center of masses of these two bodies with respect to the center of masses of the solar system. Of course \vec{b} , as well as the other magnitudes that appear in the equation that defines the transformation, is a function of time.

We denote by \vec{R}_p and \vec{R}_s the positions of the two primaries (primary and secondary) in the ecliptic system, and by \vec{R}_b the position of the barycenter. \vec{V}_s will denote the velocity of the secondary with respect to the primary.

As the barycenter is fixed at the origin in the synodic system, it follows that $\vec{b} = \vec{R}_b$.

As in the adimensional system the primaries are fixed at the points $(\mu, 0, 0)$ and $(\mu - 1, 0, 0)$, where μ is the mass parameter, it follows that $k = |\vec{R}_p - \vec{R}_s|$.

Let $\vec{c}_1, \vec{c}_2, \vec{c}_3$ denote the three column vectors of the matrix C . We have taken

$$\begin{aligned}\vec{c}_1 &= \frac{\vec{R}_p - \vec{R}_s}{k}, \\ \vec{c}_3 &= \frac{(\vec{R}_s - \vec{R}_p) \wedge \vec{V}_s}{|(\vec{R}_s - \vec{R}_p) \wedge \vec{V}_s|}, \\ \vec{c}_2 &= \vec{c}_3 \wedge \vec{c}_1.\end{aligned}$$

The determination of \vec{c}_1 is trivial if we ask for the primary being at $(\mu, 0, 0)$ and the secondary at $(\mu - 1, 0, 0)$. We have selected \vec{c}_3 in such a way that the (x, y) adimensional plane contains at every moment both primaries and the velocity of the secondary with respect to the primary.

The transformations for the velocity and the acceleration are obtained from the previous one by derivation. It is clear that the transformation of positions, velocities and accelerations requires the knowledge of velocity, acceleration and overacceleration of the secondary with respect to the primary, respectively.

The second coordinate system is the normalized system. It is chosen in such a way that the equations of motion reduce to Richardson's formulation when the perturbations are skipped. We define points in the adimensional (or synodic) system by the same geometrical relations that determine the libration points in the RTBP, despite no libration points exist in the perturbed system.

Let L_i be one of these points. To get the normalized system from the adimensional one we shift the origin of coordinates at L_i (parallel translation) and choose an unit of distance such that the distance from the primary (the secondary) to L_3

(L_1 or L_2) becomes the unity. For the L_1 and L_2 cases the positive x - and y -axes are changed of sign. For L_4 and L_5 we merely perform a translation and no scaling is required.

Now we go to the change of time. Both in the adimensional and the normalized systems the unit of time is the same one. It is defined in such a way that the mean time that the secondary spends to perform a revolution around the primary equals 2π in the new units. In the Earth+Moon-Sun system this means that the mean longitude of the Sun becomes equal to one.

At this point a problem of coherence appears. We wish to express the equations of motion as a perturbation of the RTBP. If we consider the primaries, Sun and Earth-Moon barycenter, Kepler's relation $n^2 a^3 = G(m_p + m_s)$ should hold. In this relation n is the mean motion of the related mean longitude and a is the mean distance between the primaries. It turns out that this relation is not satisfied. Then we have modified the mass of the more massive primary, m_p , to recover Kepler's third law. The remaining part of the mass (and, when required, the radiation pressure effect) is considered as a perturbation.

B.2.2 Equations of Motion

In this section we shall develop the equations of motion in the real solar system in a suitable way in order to study perturbed motions around "geometrically" defined equilibrium points related to the ones of the corresponding RTBP.

First of all we shall use ideas of Richardson and Farquhar to obtain the equations of motion in the vicinity of the three collinear points, in the Earth+Moon-Sun problem, in a general form so that:

- (1) They are readily adapted to the perturbation analysis.
- (2) All the orders of the non-linear developments are easily obtained using recursive relationships. So, the form of these equations is good for being used in the computer.
- (3) Furthermore, the equations shall be of the form obtained by Richardson, for the restricted circular three-body problem, plus perturbations.

Point 1) was solved by Farquhar without taking into account point 2). Richardson solved point 2) for the circular restricted problem omitting point 1).

We study the motion of a spacecraft in the solar system formed by the Sun, the Earth, the Moon and several planets P_1, \dots, P_k .

First, we consider an inertial frame of reference with origin at the center of masses of the solar system, whose axes are parallel to the ecliptic ones, the time t is given in modified Julian days and the distances are given in km.

Let $\vec{R}, \vec{R}_S, \vec{R}_E, \vec{R}_M, \vec{R}_B$ and \vec{R}_{P_i} (resp. O, S, E, M, B, P_i) be the position vectors with respect to the inertial frame (resp. the masses) of the spacecraft, Sun, Earth, Moon, Earth-Moon barycenter and the i th planet, respectively. The

equations of motion of the spacecraft can be written as

$$\ddot{\vec{R}} = G \sum_{A \in \{S, E, M, P_1, \dots, P_k\}} \frac{A(\vec{R}_A - \vec{R})}{|\vec{R}_A - \vec{R}|^3},$$

where G is the gravitational constant and dots denote derivatives with respect to t .

Let α be the radiation pressure of the Sun in Newton/meter² at one Astronomical Unit, i. e.: $\alpha = 4.51 \cdot 10^{-6}$. We denote by m (in kg) and σ (in square meters) the mass and the effective cross section of the spacecraft to the radiation pressure. Then, the contribution of the radiation pressure to the equations of motion is

$$-\tilde{\alpha} \frac{\vec{R}_S - \vec{R}}{|\vec{R}_S - \vec{R}|^3},$$

where

$$\tilde{\alpha} = \frac{149\,597\,870.66^2 \times 86\,400^2 \alpha \sigma}{1\,000m} \frac{\text{km}^3}{(\text{modified Julian day})^2}.$$

We denote by a_B and n_B the semimajor axis and the mean motion of the Earth–Moon barycenter around the Sun. The mean motion takes into account the secular terms in the ecliptic argument of perihelion and in the mean anomaly, due to the effect of the planets. Then, using Kepler's third law we have that $n_B^2 a_B^3$ is close to $G(S + E + M)$. We introduce a modified mass of the Sun, \bar{S} , to satisfy Kepler's third law by means of

$$n_B^2 a_B^3 = G(\bar{S} + E + M).$$

In short, taking into account the radiation pressure we can write the equations of motion in the following form

$$\ddot{\vec{R}} = \frac{G\bar{S}(\vec{R}_S - \vec{R})}{|\vec{R}_S - \vec{R}|^3} + \frac{G\tilde{S}(\vec{R}_S - \vec{R})}{|\vec{R}_S - \vec{R}|^3} + \sum_{A \in \{E, M, P_1, \dots, P_k\}} \frac{GA(\vec{R}_A - \vec{R})}{|\vec{R}_A - \vec{R}|^3},$$

where $\tilde{S} = S - \bar{S} - \tilde{\alpha}/G$. The contribution of $S - \bar{S}$ to \tilde{S} in the above equations is called the coherence term (or Kepler coherence term).

The position of the Sun satisfies

$$\ddot{\vec{R}}_S = \sum_{A \in \{E, M, P_1, \dots, P_k\}} \frac{GA(\vec{R}_A - \vec{R}_S)}{|\vec{R}_A - \vec{R}_S|^3}.$$

So, if $\vec{r} = \vec{R} - \vec{R}_S$, then the equations of motion of the spacecraft with respect to

the ecliptic frame of reference with origin at the Sun are

$$\ddot{\vec{r}} = -\frac{G\bar{S}\vec{r}}{|\vec{r}|^3} - \frac{G\tilde{S}\vec{r}}{|\vec{r}|^3} + \sum_{A \in \{E, M, P_1, \dots, P_k\}} GA \left(\frac{\vec{r}_{SA} - \vec{r}}{|\vec{r}_{SA} - \vec{r}|^3} - \frac{\vec{r}_{SA}}{|\vec{r}_{SA}|^3} \right),$$

where \vec{r}_{SA} denotes the vector from S to A .

We note that the above equations correspond to the Lagrange equations given by the Lagrangian

$$L = \frac{1}{2} \dot{\vec{r}} \cdot \dot{\vec{r}} + \frac{G\bar{S}}{|\vec{r}|} + \frac{G\tilde{S}}{|\vec{r}|} + \sum_{A \in \{E, M, P_1, \dots, P_k\}} GA \left(\frac{1}{|\vec{r}_{SA} - \vec{r}|} - \frac{\vec{r}_{SA} \cdot \vec{r}}{|\vec{r}_{SA}|^3} \right),$$

or equivalently

$$\begin{aligned} L = & \frac{1}{2} \dot{\vec{r}} \cdot \dot{\vec{r}} + K \left(\frac{1 - \mu_B}{|\vec{r}|} + \frac{\mu_{\tilde{S}}}{|\vec{r}|} \right) \\ & + K \sum_{A \in \{E, M, P_1, \dots, P_k\}} \mu_A \left(\frac{1}{|\vec{r}_{SA} - \vec{r}|} - \frac{\vec{r}_{SA} \cdot \vec{r}}{|\vec{r}_{SA}|^3} \right), \end{aligned}$$

where $K = G(\bar{S} + E + M)$ and $\mu_A = A(\bar{S} + E + M)^{-1}$ for $A \in \{\bar{S}, \tilde{S}, E, M, P_1, \dots, P_k\}$. Note that $\mu_{\bar{S}} = 1 - \mu_B$ where $\mu_B = (E + M)(\bar{S} + E + M)^{-1}$.

The collinear L_1 point spacecraft geometry for the Sun and the Earth+Moon barycenter is illustrated in Figure B.4.

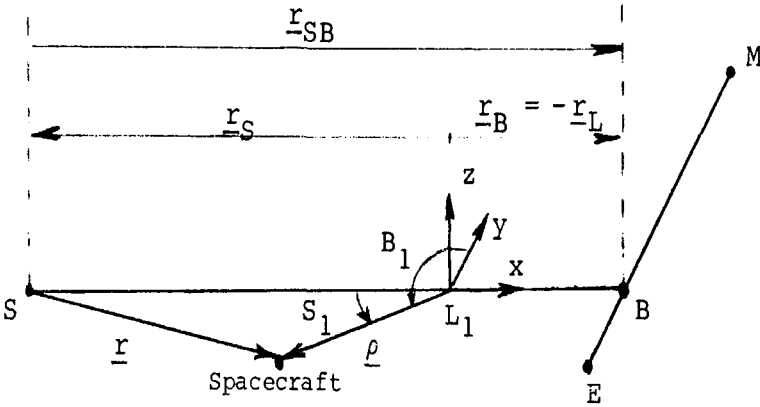


Fig. B.4 Collinear L_1 point spacecraft geometry for the Sun and the Earth+Moon barycenter.

The geometry for the other equilibria is similar and follows the one defined for the RTBP. Let L_i (for $i = 1, 2, 3$) be located with respect to the Earth+Moon

barycenter by the vector \vec{r}_L . The expression determining \vec{r}_L is

$$\frac{\vec{r}_L}{|\vec{r}_{SB}|^3} - \frac{\mu_B \vec{r}_L}{|\vec{r}_L|^3} = (1 - \mu_B) \left(\frac{\vec{r}_S}{|\vec{r}_S|^3} - \frac{\vec{r}_{SB}}{|\vec{r}_{SB}|^3} \right).$$

This equation is equivalent to the quintic of Euler for the determination of the collinear points of the circular restricted problem. Let $\gamma_1 = r_{L_1} r_{SB}^{-1}$, $\gamma_2 = r_{L_2} r_{SB}^{-1}$, $\gamma_3 = r_{L_3} r_{SB}^{-1}$. So the locations of the collinear points L_i are obtained from the solutions of the quintics:

$$\gamma_1^5 - (3 - \mu_B) \gamma_1^4 + (3 - 2\mu_B) \gamma_1^3 - \mu_B \gamma_1^2 + 2\mu_B \gamma_1 - \mu_B = 0,$$

$$\gamma_2^5 + (3 - \mu_B) \gamma_2^4 + (3 - 2\mu_B) \gamma_2^3 - \mu_B \gamma_2^2 - 2\mu_B \gamma_2 - \mu_B = 0,$$

$$\gamma_3^5 + (2 + \mu_B) \gamma_3^4 + (1 + 2\mu_B) \gamma_3^3 - (1 - \mu_B) \gamma_3^2 - 2(1 - \mu_B) \mu_B \gamma_3 - (1 - \mu_B) = 0.$$

Now we perform a translation of our reference system to the point L_i of the Earth+Moon-Sun system, i.e. let $\vec{\rho} = \vec{r} - \vec{r}_{SL}$ be the position vector of the spacecraft in the ecliptic frame of reference with origin at L_i . Positions of the Sun, Earth, Moon, a planet P_i and the Earth-Moon barycenter with respect to L_i are denoted by \vec{r}_S , \vec{r}_E , \vec{r}_M , \vec{r}_{P_i} , and \vec{r}_B respectively.

In this new frame of reference the Lagrangian goes over to

$$\begin{aligned} L = & \frac{1}{2} \dot{\vec{\rho}} \cdot \dot{\vec{\rho}} - \dot{\vec{\rho}} \cdot \dot{\vec{r}}_S + K \left(\frac{1 - \mu_B}{|\vec{r}_S - \vec{\rho}|} + \mu_B \left(\frac{1}{|\vec{r}_B - \vec{\rho}|} - \frac{\vec{r}_{SB} \cdot \vec{\rho}}{|\vec{r}_{SB}|^3} \right) \right) \\ & + \frac{K \mu_{\tilde{S}}}{|\vec{r}_S - \vec{\rho}|} + K \sum_{A \in \{E, M, P_1, \dots, P_k, B\}} i(A) \mu_A \left(\frac{1}{|\vec{r}_A - \vec{\rho}|} - \frac{\vec{r}_{SA} \cdot \vec{\rho}}{|\vec{r}_{SA}|^3} \right), \end{aligned}$$

where we omit the terms in the Lagrangian which only depend on time and

$$i(A) = \begin{cases} 1 & \text{if } A \in \{E, M, P_1, \dots, P_k\}, \\ -1 & \text{if } A = B. \end{cases}$$

The Lagrangian can be written in the form

$$\begin{aligned} L = & \frac{1}{2} \dot{\vec{\rho}} \cdot \dot{\vec{\rho}} - \dot{\vec{\rho}} \cdot \dot{\vec{r}}_S + K(1 - \mu_B) \left(\frac{1}{|\vec{r}_S - \vec{\rho}|} - \frac{\vec{r}_S \cdot \vec{\rho}}{|\vec{r}_S|^3} \right) \\ & + K \mu_B \left(\frac{1}{|\vec{r}_B - \vec{\rho}|} - \frac{\vec{r}_B \cdot \vec{\rho}}{|\vec{r}_B|^3} \right) \\ & + K \left((1 - \mu_B) \frac{\vec{r}_S \cdot \vec{\rho}}{|\vec{r}_S|^3} + \mu_B \left(\frac{\vec{r}_B \cdot \vec{\rho}}{|\vec{r}_B|^3} - \frac{\vec{r}_{SB} \cdot \vec{\rho}}{|\vec{r}_{SB}|^3} \right) \right) \\ & + \frac{K \mu_{\tilde{S}}}{|\vec{r}_S - \vec{\rho}|} \\ & + K \sum_{A \in \{E, M, P_1, \dots, P_k, B\}} i(A) \mu_A \left(\frac{1}{|\vec{r}_A - \vec{\rho}|} - \frac{\vec{r}_{SA} \cdot \vec{\rho}}{|\vec{r}_{SA}|^3} \right). \end{aligned}$$

Using the definition of L_i we have

$$(1 - \mu_B) \frac{\vec{r}_S}{|\vec{r}_S|^3} + \mu_B \frac{\vec{r}_B}{|\vec{r}_B|^3} = \frac{\vec{r}_B - (1 - \mu_B)\vec{r}_{SB}}{|\vec{r}_{SB}|^3}.$$

So,

$$\begin{aligned} & (1 - \mu_B) \frac{\vec{r}_S \cdot \vec{\rho}}{|\vec{r}_S|^3} + \mu_B \left(\frac{\vec{r}_B \cdot \vec{\rho}}{|\vec{r}_B|^3} - \frac{\vec{r}_{SB} \cdot \vec{\rho}}{|\vec{r}_{SB}|^3} \right) \\ &= \left(\frac{\vec{r}_B - (1 - \mu_B)\vec{r}_{SB}}{|\vec{r}_{SB}|^3} - \frac{\mu_B \vec{r}_{SB}}{|\vec{r}_S|^3} \right) \cdot \vec{\rho} = \frac{\vec{r}_S \cdot \vec{\rho}}{|\vec{r}_{SB}|^3}. \end{aligned}$$

Hence the Lagrangian becomes

$$\begin{aligned} L = & \frac{1}{2} \dot{\vec{\rho}} \cdot \dot{\vec{\rho}} - \dot{\vec{\rho}} \cdot \dot{\vec{r}}_S + K(1 - \mu_B) \left(\frac{1}{|\vec{r}_S - \vec{\rho}|} - \frac{\vec{r}_S \cdot \vec{\rho}}{|\vec{r}_S|^3} \right) \\ & + K\mu_B \left(\frac{1}{|\vec{r}_B - \vec{\rho}|} - \frac{\vec{r}_B \cdot \vec{\rho}}{|\vec{r}_B|^3} \right) \\ & + \frac{K\vec{r}_S \cdot \vec{\rho}}{|\vec{r}_{SB}|^3} + \frac{K\mu_S}{|\vec{r}_S - \vec{\rho}|} \\ & + K \sum_{A \in \{E, M, P_1, \dots, P_k, B\}} i(A) \mu_A \left(\frac{1}{|\vec{r}_A - \vec{\rho}|} - \frac{\vec{r}_{SA} \cdot \vec{\rho}}{|\vec{r}_{SA}|^3} \right). \end{aligned}$$

The next step is to express the Lagrangian in normalized coordinates.

First, we consider a frame of reference at the equilibrium point L_i , the x -axis points along the line formed by the two primaries (the Sun and the Earth–Moon barycenter) away from the larger primary, the (x, y) -plane coincides with the instantaneous plane of motion of the primaries, and the z -axis completes the right-handed system.

Let θ and δ be the longitude and the latitude of the Earth–Moon barycenter in ecliptic coordinates centered at the Sun. Then, the unitary vector on the x -axis is

$$\vec{e}_1 = \frac{\vec{r}_{SB}}{|\vec{r}_{SB}|},$$

the unitary vector on the z -axis

$$\vec{e}_3 = \frac{\vec{r}_{SB} \wedge \dot{\vec{r}}_{SB}}{|\vec{r}_{SB} \wedge \dot{\vec{r}}_{SB}|},$$

and the unitary vector on the y -axis is $\vec{e}_2 = \vec{e}_3 \wedge \vec{e}_1$. Let C be the matrix whose columns are the three vectors $\vec{e}_1, \vec{e}_2, \vec{e}_3$:

$$C = \begin{pmatrix} \cos \delta \cos \theta & -T(\sin \theta + R \sin \delta \cos \theta) & T(-\cos \theta \sin \delta + R \sin \theta) \\ \cos \delta \sin \theta & T(\cos \theta - R \sin \delta \sin \theta) & -T(\sin \theta \sin \delta + R \cos \theta) \\ \sin \delta & TR \cos \delta & T \cos \delta \end{pmatrix},$$

where $R = \dot{\delta}(\dot{\theta} \cos \delta)^{-1}$ and $T = (1 + R^2)^{-1/2}$. Note that the matrix C is orthogonal.

Now we consider the change of variables from the ecliptic frame of reference with origin at L_i , $\vec{\rho}$, to the frame of reference with origin at L_i , $\vec{a} = (x, y, z)$, given by

$$\vec{\rho} = kC\vec{a},$$

where the scaling factor is taken as $k = r_{L_i} = \gamma_i r_{SB}$ for $L_{1,2}$ and as $k = r_S = \gamma_i r_{SB}$ for L_3 .

In the system (x, y, z) units of mass, time and distance are chosen in the following way (here γ_i is replaced by γ for shortness):

- The unit of time is taken as γ times the semimajor axis a_B .
- The new unit of time s is the time t in modified Julian days scaled by $\gamma^{3/2}$, so $s = \gamma^{3/2}t$. In what follows primes will denote derivatives with respect to s .
- The unit of mass is chosen in such a way that

$$K = G(\bar{S} + E + M) = n_B^2 a_B^3 = 1,$$

where n_B is the mean motion.

System (x, y, z) with these new units is the normalized system centered at L_i . We remark that now the passage from ecliptic to normalized coordinates is done performing first the translation to L_i .

Taking into account the properties of the matrix C and developing the last terms in the Lagrangian by means of Legendre polynomials, the new Lagrangian becomes:

$$\begin{aligned} L = & \frac{1}{2} \{ k^2(x'^2 + y'^2 + z'^2) + 2kk'(xx' + yy' + zz') \\ & + 2k^2(E(xy' - yx') + F(yz' - zy')) + k'^2(x^2 + y^2 + z^2) \\ & + k^2(Ax^2 + By^2 + Cz^2 + 2Dxz) \} \\ & - k'^2 x_S x - kk' x_S x' - k^2(Ax_S x + Dx_S z) - k^2 E x_E y' \\ & + Kk^{-1} \sum_{n \geq 2} c_n a^n P_n\left(\frac{x}{a}\right) + Kk^{-1} x_S x \\ & + Kk^{-1} \gamma^{-3} \frac{\mu \tilde{S}}{\tilde{r}_S} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_S}\right)^n P_n(\cos S_1) \\ & + Kk^{-1} \gamma^{-3} \sum_{A \in \mathcal{P}} i(A) \mu_A \left(-\frac{a \cos A_2}{\tilde{r}_{SA}^2} + \frac{1}{\tilde{r}_A} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_A}\right)^n P_n(\cos A_1) \right), \end{aligned}$$

where $\mathcal{P} = \{E, M, P_i, B\}$ and the normalized vectors have been denoted with a tilde. We recall that in normalized coordinates \vec{r}_E and \vec{r}_S have only x components. The angle A_1 (resp. A_2) is the angle between \vec{r}_A (resp. \vec{r}_{SA}) and a .

The coefficients A, B, C, D, E , and F , as well as its derivatives, are functions of the longitude θ and the latitude δ of the Earth-Moon barycenter and are defined

by

$$\begin{aligned}
 A &= \theta'^2 \cos^2 \delta + \delta'^2, \\
 B &= \frac{(\cos^2 \delta + \sin^2 \delta R^2) \theta'^2 + (2 + R^2) \delta'^2 + (R' + \sin \delta \theta')^2}{(1 + R^2)} \\
 &\quad - \frac{R^2 R'^2}{(1 + R^2)^2}, \\
 C &= \frac{R^2 \theta'^2 - \delta'^2 + (R' + \sin \delta \theta')^2}{(1 + R^2)} - \frac{R^2 R'^2}{(1 + R^2)^2}, \\
 D &= -\sin \delta \cos \delta \theta'^2 (1 + R^2)^{1/2} - \frac{\cos \delta \theta' R'}{(1 + R^2)^{1/2}}, \\
 E &= \frac{\theta' \cos \delta + \delta' R}{(1 + R^2)^{1/2}}, \\
 F &= \theta' \sin \delta + \frac{R'}{(1 + R^2)},
 \end{aligned}$$

where

$$R = \frac{\delta'}{\theta' \cos \delta}.$$

From the Lagrangian, the equations of motion are easily written, see [2] for the details, as:

$$\begin{aligned}
 x'' &= 2y' + (1 + 2c_2)x + \sum_{n \geq 2} (n+1)c_{n+1}a^n P_n\left(\frac{x}{a}\right) \\
 &\quad + C(0) \sum_{n \geq 2} n c_n a^{n-1} P_n\left(\frac{x}{a}\right) \\
 &\quad + C(1)x + C(2)y + C(3)z + C(4)x' + C(5)y' + C(7) \\
 &\quad + K k^{-3} \gamma^{-3} \mu_{\tilde{S}}(x - x_S) \sum_{n \geq 2} \frac{a^{n-2}}{x_S^{n+1}} \bar{P}_n(\cos S_1) \\
 &\quad + K k^{-3} \gamma^{-3} \sum_{A \in \mathcal{P}} i(A) \mu_A \left(-\frac{x_{SA}}{\tilde{r}_{SA}^3} + (x - x_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \bar{P}_n(\cos A_1) \right), \\
 y'' &= -2x' + (1 - c_2)y + y \sum_{n \geq 3} c_n a^{n-2} \bar{P}_n\left(\frac{x}{a}\right) \\
 &\quad + C(0)y \sum_{n \geq 2} c_n a^{n-2} \bar{P}_n\left(\frac{x}{a}\right) \\
 &\quad + C(11)x + C(12)y + C(13)z + C(14)x' + C(15)y' + C(16)z' \\
 &\quad + C(17) + K k^{-3} \gamma^{-3} \mu_{\tilde{S}} y \sum_{n \geq 2} \frac{a^{n-2}}{x_S^{n+1}} \bar{P}_n(\cos S_1) \\
 &\quad + K k^{-3} \gamma^{-3} \sum_{A \in \mathcal{P}} i(A) \mu_A \left(-\frac{y_{SA}}{\tilde{r}_{SA}^3} + (y - y_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \bar{P}_n(\cos A_1) \right),
 \end{aligned}$$

$$\begin{aligned}
z'' &= -c_2 z + z \sum_{n \geq 3} c_n a^{n-2} \bar{P}_n \left(\frac{x}{a} \right) \\
&+ C(0) z \sum_{n \geq 2} c_n a^{n-2} \bar{P}_n \left(\frac{x}{a} \right) \\
&+ C(21)x + C(22)y + C(23)z + C(25)y' + C(26)z' + C(27) \\
&+ K k^{-3} \gamma^{-3} \mu_{\tilde{S}} z \sum_{n \geq 2} \frac{a^{n-2}}{x_S^{n+1}} \bar{P}_n(\cos S_1) \\
&+ K k^{-3} \gamma^{-3} \sum_{A \in \mathcal{P}} i(A) \mu_A \left(-\frac{z_{SA}}{\tilde{r}_{SA}^3} + (z - z_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \bar{P}_n(\cos A_1) \right),
\end{aligned}$$

where

$$\mathcal{P} = \{S, M, P_1, \dots, P_k\}, \quad \bar{P}_n(\alpha) = -d(P_{n-1}(\alpha))/d\alpha$$

and

$$\begin{aligned}
C(0) &= K k^{-3} - 1, \\
C(1) &= A - 1 - k'' k^{-1}, \\
C(2) &= 2k' k^{-1} E + E', \\
C(3) &= D, \\
C(4) &= -2k' k^{-1}, \\
C(5) &= 2(\theta' - \theta'^2 + E - 1), \\
C(7) &= (k'' k^{-1} - A + K k^{-3}) x_S, \\
C(12) &= B - 1 - k'' k^{-1}, \\
C(13) &= 2k' k^{-1} F + F', \\
C(16) &= 2F, \\
C(23) &= C - k'' k^{-1},
\end{aligned}$$

being $C(11) = -C(2)$, $C(14) = -C(5)$, $C(15) = C(4)$, $C(17) = x_S C(2)$, $C(21) = C(3)$, $C(22) = -C(13)$, $C(25) = -C(16)$, $C(26) = C(4)$, $C(27) = -x_S C(3)$.

Appendix C

The Model Equations Near the Equilateral Points in the Earth–Moon System

C.1 Introduction

The equations of the real motion near L_4 or L_5 in the Earth–Moon system are written. The full solar system and the radiation pressure are considered. The Hamiltonian of the problem is also given. The terms of the Lagrangian and the Hamiltonian which involve Legendre polynomials are expanded as power series in x , y and z with coefficients which are periodic functions of time (with incommensurable frequencies). A Fourier analysis of such functions as well as the other functions needed for the equations and Hamiltonian is performed.

The number of terms to be retained in the expansions of the Hamiltonian and the equations, with some tolerance, is given. This suggests to write down the equations of motion in a simplified way. The full coefficients which appear in the equations finally used are given. Checks against the full vector field are done.

C.2 Systems of Reference

We consider an inertial frame of reference with the origin at the center of masses of the solar system and the axes parallel to the ecliptic ones. The equations of motion of a spacecraft in the solar system can be written as

$$\ddot{\vec{R}} = G \sum_{A \in \{S, E, P_1, \dots, P_k\}} \frac{A(\vec{R}_A - \vec{R})}{|\vec{R}_A - \vec{R}|^3}$$

as we did in B.2.2 . In what follows we proceed in a way similar to the collinear case presented in B.2 with the suitable changes.

However, the above system of reference is not convenient to study the motion of a spacecraft in the vicinity of the libration points L_4 or L_5 corresponding to the Earth–Moon system. As it is usual, the libration points are defined as the ones that form an equilateral triangle with the Earth and the Moon. These points are

placed at the instantaneous plane of motion of the Moon around the Earth (see Figure C.1)

We define a normalized reference system centered at the libration points and given by the unitary vectors \vec{e}_1 , \vec{e}_2 and \vec{e}_3 :

$$\begin{aligned}\vec{e}_1 &= \frac{\vec{r}_{EM}}{|\vec{r}_{EM}|}, \\ \vec{e}_3 &= \frac{\vec{r}_{EM} \wedge \vec{r}_{EM}}{|\vec{r}_{EM} \wedge \vec{r}_{EM}|}, \\ \vec{e}_2 &= \vec{e}_3 \wedge \vec{e}_1.\end{aligned}$$

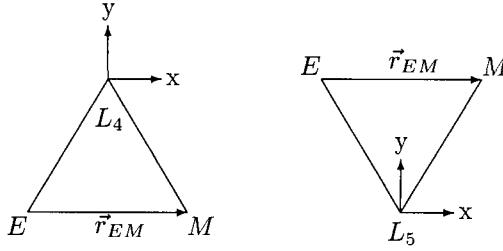


Fig. C.1 Reference systems at the triangular equilibrium points.

We introduce a modified mass of the Earth, \bar{E} , to satisfy the third Kepler's law

$$K = G(\bar{E} + M) = n_M^2 a_M^3$$

where n_M and a_M are the mean motion and the semimajor axis of the Moon around the Earth. The rest of mass $\tilde{E} = E - \bar{E}$ will be considered as a perturbation. Then, in this normalized system (x, y, z) , we adopt the following units:

- The unit of mass is chosen such that $G(\bar{E} + M) = 1$,
- The unit of time is defined to have $n_M = 1$, The unit of distance is taken as r_{EM} .

Therefore, in the L_4 case, the coordinates of the Earth and the Moon are:

- $(x_E, y_E, z_E) = (-\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0)$,
- $(x_M, y_M, z_M) = (\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0)$.

In a similar way, for L_5 we have:

- $(x_E, y_E, z_E) = (-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0)$,
- $(x_M, y_M, z_M) = (\frac{1}{2}, \frac{\sqrt{3}}{2}, 0)$.

For a precise definition of the adopted system see [3].

C.3 The Lagrangian

In the normalized system of reference centered at the libration point L_i , $i = 4, 5$, the Lagrangian (see [3]) can be expanded in Legendre polynomials as:

$$\begin{aligned}
 L = & \frac{1}{2} \{ k^2 (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + 2k\dot{k}(x\dot{x} + y\dot{y} + z\dot{z}) \\
 & + 2k^2 (E(x\dot{y} - y\dot{x}) + F(y\dot{z} - z\dot{y})) + \dot{k}^2 (x^2 + y^2 + z^2) \\
 & + k^2 (Ax^2 + By^2 + Cz^2 + 2Dxz) \} \\
 & - \dot{k}^2 (x_E x + y_E y) - k\dot{k}(x_E \dot{x} + y_E \dot{y}) - k^2 (Ax_E x + By_E y + Dz_E z) \\
 & - k^2 (-Ey_E \dot{x} + Ex_E \dot{y} + Fy_E \dot{z}) \\
 & + Kk^{-1} (1 - \mu_M + \mu_E) \sum_{n \geq 1} a^n P_n(\cos E_1) \\
 & + Kk^{-1} \frac{\mu_{\tilde{S}}}{\tilde{r}_S} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_S} \right)^n P_n(\cos S_1) \\
 & + Kk^{-1} \sum_{A \in \{S, M, P_1, \dots, P_k\}} \mu_A \left(-\frac{a \cos A_2}{\tilde{r}_{EA}^2} + \frac{1}{\tilde{r}_A} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_A} \right)^n P_n(\cos A_1) \right),
 \end{aligned}$$

where $\vec{a} = (x, y, z)$ is the position vector of the spacecraft, \vec{v} stands for the vector \vec{v} in the normalized system, $k = r_{EM}$ is a scaling factor, $\cos A_1 = (\vec{\tilde{r}}_A, \vec{a})/(\tilde{r}_A a)$ and $\cos A_2 = (\vec{\tilde{r}}_{EA}, \vec{a})/(\tilde{r}_{EA} a)$. We denote by V the modulus of the vector \vec{V} .

The term with $\mu_{\tilde{S}}$ takes into account the radiation pressure of the Sun.

The coefficients A, B, C, D, E, and F are the ones given in B.2.2, replacing θ' , δ' by $\dot{\theta}$, $\dot{\delta}$ and using now the geocentric longitude, θ , and latitude, δ , of the Moon.

From the Lagrangian, the equations of motion are easily written as:

$$\begin{aligned}
 \ddot{x} - 2\dot{y} - x = & C(1)x + C(2)y + C(3)z \\
 & + C(4)\dot{x} + C(5)\dot{y} + C(7) \\
 & + Kk^{-3} (1 - \mu_M + \mu_E) (x - x_E) \sum_{n \geq 2} a^{n-2} \bar{P}_n(\cos E_1) \\
 & + Kk^{-3} \mu_{\tilde{S}} (x - x_S) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_S^{n+1}} \bar{P}_n(\cos S_1) \\
 & + Kk^{-3} \sum_{A \in \mathcal{P}} \mu_A \left(-\frac{x_{EA}}{\tilde{r}_{EA}^3} + (x - x_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \bar{P}_n(\cos A_1) \right),
 \end{aligned}$$

$$\begin{aligned}
\ddot{y} + 2\dot{x} - y &= C(11)x + C(12)y + C(13)z \\
&\quad + C(14)\dot{x} + C(15)\dot{y} + C(16)\dot{z} + C(17) \\
&\quad + Kk^{-3}(1 - \mu_M + \mu_{\bar{E}})(y - y_E) \sum_{n \geq 2} a^{n-2} \bar{P}_n(\cos E_1) \\
&\quad + Kk^{-3} \mu_{\bar{S}}(y - y_S) \sum_{n \geq 2} \frac{a^{n-2}}{\bar{r}_S^{n+1}} \bar{P}_n(\cos S_1) \\
&\quad + Kk^{-3} \sum_{A \in \mathcal{P}} \mu_A \left(-\frac{y_{EA}}{\bar{r}_{EA}^3} + (y - y_A) \sum_{n \geq 2} \frac{a^{n-2}}{\bar{r}_A^{n+1}} \bar{P}_n(\cos A_1) \right), \\
\ddot{z} &= C(21)x + C(22)y + C(23)z \\
&\quad + C(25)\dot{y} + C(26)\dot{z} + C(27) \\
&\quad + Kk^{-3}(1 - \mu_M + \mu_{\bar{E}})z \sum_{n \geq 2} a^{n-2} \bar{P}_n(\cos E_1) \\
&\quad + Kk^{-3} \mu_{\bar{S}}(z - z_S) \sum_{n \geq 2} \frac{a^{n-2}}{\bar{r}_S^{n+1}} \bar{P}_n(\cos S_1) \\
&\quad + Kk^{-3} \sum_{A \in \mathcal{P}} \mu_A \left(-\frac{z_{EA}}{\bar{r}_{EA}^3} + (z - z_A) \sum_{n \geq 2} \frac{a^{n-2}}{\bar{r}_A^{n+1}} \bar{P}_n(\cos A_1) \right),
\end{aligned}$$

where $\mathcal{P} = \{S, M, P_1, \dots, P_k\}$, \bar{P}_n , is as in B.2.2,

$$\begin{aligned}
C(7) &= (\ddot{k}k^{-1} - A)x_E - (2\dot{k}k^{-1}E + \dot{E})y_E, \\
C(17) &= (\ddot{k}k^{-1} - B)y_E + (2\dot{k}k^{-1}E + \dot{E})x_E, \\
C(27) &= -Dx_E + (2\dot{k}k^{-1}F + \dot{F})y_E,
\end{aligned}$$

and the other values of $C(j)$ are as in B.2.2 replacing $' (= d/ds)$ by $\dot{} (= d/dt)$.

C.4 The Hamiltonian and the Related Expansions

The Hamiltonian will not be required in what follows. However we give it because of its possible use in future work.

We start with the expression of the Lagrangian (given in [C.3]) of the problem when normalized coordinates centered at the instantaneous $L_{4,5}$ point of the Earth-Moon system are used.

The momenta p_x, p_y, p_z are introduced through

$$\begin{aligned}
p_x &= \partial L / \partial \dot{x} = k^2 \dot{x} + k\dot{k}x - k^2 E y - k\dot{k}x_E + k^2 E y_E, \\
p_y &= \partial L / \partial \dot{y} = k^2 \dot{y} + k\dot{k}y + k^2 (Ex - Fz) - k\dot{k}y_E - k^2 E x_E, \\
p_z &= \partial L / \partial \dot{z} = k^2 \dot{z} + k\dot{k}z + k^2 F y - k^2 F y_E.
\end{aligned}$$

From these relations it is easy to express $\dot{x}, \dot{y}, \dot{z}$ as functions of the positions and momenta, the coefficients being functions of time.

The Hamiltonian is obtained as

$$H = \sum p_i \dot{q}_i - L,$$

with

$$q = (x, y, z)^T, \quad p = (p_x, p_y, p_z)^T,$$

where in \dot{q}_i , as well as in $\dot{x}^2 + \dot{y}^2 + \dot{z}^2$, $x\dot{x} + y\dot{y} + z\dot{z}$, $E(x\dot{y} - y\dot{x}) + F(y\dot{z} - z\dot{y})$, $x_E\dot{x} + y_E\dot{y}$ and $-E y_E\dot{x} + E x_E\dot{y} + F y_E\dot{z}$, $\dot{x}, \dot{y}, \dot{z}$ are substituted by their expressions in terms of positions and momenta.

After a somewhat lengthy computations the following Hamiltonian is obtained

$$\begin{aligned} H = & \frac{1}{2}k^{-2}(p_x^2 + p_y^2 + p_z^2) - \dot{k}k^{-1}(xp_x + yp_y + zp_z) \\ & + E(y p_x - x p_y) + F(z p_y - y p_z) + \\ & + (\dot{k}k^{-1}x_E - E y_E)p_x + (\dot{k}k^{-1}y_E + E x_E)p_y + F y_S p_z \\ & + \text{Terms purely dependent on time} \\ & - K k^{-1}(1 - \mu_M + \mu_{\tilde{E}}) \sum_{n \geq 1} a^n P_n(\cos E_1) \\ & - K k^{-1} \frac{\mu_{\tilde{S}}}{\tilde{r}_S} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_S} \right)^n P_n(\cos S_1) \\ & + K k^{-1} \sum_{A \in \mathcal{P}} \mu_A \left(-\frac{a \cos A_2}{\tilde{r}_{EA}^2} + \frac{1}{\tilde{r}_A} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_A} \right)^n P_n(\cos A_1) \right), \end{aligned}$$

The time-dependent functions which appear as coefficients have been partly computed in the development of the Lagrangian. The only additional function is k^{-2} .

C.5 Some Useful Expansions

In order to get a quasi-periodic solution of the problem it is convenient to develop the right-hand side of the equations as series of the following types

$$\sum a_{ijk} x^i y^j z^k F(\nu_r t + \varphi_r),$$

or

$$\dot{v} \sum a_r F(\nu_r t + \varphi_r),$$

where \dot{v} denotes \dot{x}, \dot{y} or \dot{z} and F stands for one of the trigonometric functions sine or cosine.

A Fourier analysis of the functions that appear in the equations will give the coefficients, a_{ijk_r} , a_r , the frequencies, ν_r , and the phases, φ_r , involved in the dominant terms.

We deal separately with the terms of the equations which involve the Legendre polynomials. These terms come through derivation with respect to x , y or z , of the corresponding terms of the Lagrangian. So it is more convenient to develop the summations of the Lagrangian and then to compute the derivatives.

Using the normalized system of coordinates we have

$$\cos E_1 = \frac{xx_E + yy_E + zz_E}{a\tilde{r}_E} = -\frac{1}{2a}(x + s\sqrt{3}y),$$

$$\cos M_1 = \frac{xx_M + yy_M + zz_M}{a\tilde{r}_M} = \frac{1}{2a}(x - s\sqrt{3}y),$$

with $s = 1$ for L_4 and -1 for L_5 .

Therefore, it is not difficult to see that

$$\begin{aligned} \sum_{n \geq 1} a^n P_n(\cos A_1) &= \sum_{n \geq 1} \frac{1}{2^{2n}} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^{k+nJ}}{(n-k)!} 2^{2k} (2(n-k))! \\ &\times \sum_{m=0}^{n-2k} \frac{(r_s \sqrt{3})^m}{m!(n-2k-m)!} \sum_{n_1+n_2+n_3=k} \frac{1}{n_1!n_2!n_3!} x^{2n_1+n-2k-m} y^{2n_2+m} z^{2n_3} \end{aligned}$$

where

$$J = \begin{cases} 1 & \text{if } A = E, \\ 0 & \text{if } A = M, \end{cases}$$

and

$$r = \begin{cases} 1 & \text{if } A = E, \\ -1 & \text{if } A = M. \end{cases}$$

A routine, EXPEAT, gives the coefficients of each monomial $x^i y^j z^k$ up to a given order $n = i + j + k$. This routine has been checked comparing the development with the generating function

$$\frac{1}{\sqrt{1 - 2a \cos A_1 + a^2}} - 1,$$

where $A = E$ or M . It has been seen that to get a difference of the order of 10^{-5} , terms up to order 20 are required if the magnitude of a is 0.5, but only order 8 is required if $a = 0.15$.

We note that the coefficients in the development of

$$\sum_{n \geq 1} a^n P_n(\cos A_1),$$

for $A = E$ or M are constants and so no Fourier analysis is needed for those terms.

For the planets and the Sun, similar expansions are obtained where the coefficients of the monomials are functions of time. We have:

- For the planets

$$Kk^{-1}\mu_A \frac{1}{\tilde{r}_A} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_A} \right)^n P_n(\cos A_1).$$

- For the Sun

$$Kk^{-1}(\mu_{\tilde{S}} + \mu_S) \frac{1}{\tilde{r}_S} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_S} \right)^n P_n(\cos S_1).$$

In general, we have the following expansion

$$\begin{aligned} \frac{1}{\tilde{r}_A} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_A} \right)^n P_n(\cos A_1) &= \sum_{n \geq 1} \frac{1}{2^n} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{(2(n-k))!}{(n-k)!} \\ &\times \sum_{n_1=0}^k \sum_{n_2=0}^{k-n_1} \frac{1}{n_1! n_2! (k-n_1-n_2)!} \sum_{l_1=0}^{n-2k} \sum_{l_2=0}^{n-2k-l_1} \frac{1}{l_1! l_2! (n-2k-l_1-l_2)!} \\ &\times \left(\frac{x_A}{\tilde{r}_A} \right)^{l_1} \left(\frac{y_A}{\tilde{r}_A} \right)^{l_2} \left(\frac{z_A}{\tilde{r}_A} \right)^{l_3} \left(\frac{1}{\tilde{r}_A} \right)^{n+1} x^{2n_1+l_1} y^{2n_2+l_2} z^{2n_3+l_3}, \end{aligned}$$

where $l_3 = n - 2k - l_1 - l_2$.

The terms of the expansion can be collected in the form

$$\sum_{q_1, q_2, q_3} f(q_1, q_2, q_3) x^{q_1} y^{q_2} z^{q_3},$$

where

$$f(q_1, q_2, q_3) = \sum_{l_1, l_2, l_3} c_{q_1 q_2 q_3 l_1 l_2 l_3} \left(\frac{x_A}{\tilde{r}_A} \right)^{l_1} \left(\frac{y_A}{\tilde{r}_A} \right)^{l_2} \left(\frac{z_A}{\tilde{r}_A} \right)^{l_3} \left(\frac{1}{\tilde{r}_A} \right)^{q_1+q_2+q_3+1}$$

The coefficients, $c_{q_1 q_2 q_3 l_1 l_2 l_3}$ and the exponents, $q_1, q_2, q_3, l_1, l_2, l_3$, are computed up to a given order $n = q_1 + q_2 + q_3$ in the routine EXPEAT. The routine has been checked using the generating function

$$\frac{1}{\sqrt{\tilde{r}_A^2 - 2a\tilde{r}_A \cos A_1 + a^2}} - \frac{1}{\tilde{r}_A}.$$

Some computations for the Sun, Venus and Jupiter show that order 3 is needed for the Sun, and order 2 is sufficient for the planets in order to have a difference of order 10^{-5} .

A routine, FUN4, has been done to compute the time-dependent functions at a given day. This routine will be used by the program that performs the Fourier analysis.

C.6 Fourier Analysis: The Relevant Frequencies and the Related Coefficients.

First of all the equations of motion as given before, are rewritten in the more convenient form displayed below:

$$\begin{aligned}
 \ddot{x} - 2\dot{y} - 3x_E^2x - 3(1 - 2\mu_M)x_E y_E y &= \overline{C(1)}x + \overline{C(2)}y + C(3)z + C(4)\dot{x} + C(5)\dot{y} + \overline{C(7)} \\
 + Kk^{-3}[-3(1 - 2m_M)x_E x^2 - 3y_E xy] &+ Kk^{-3}(1 - \mu_M)(x - x_E) \sum_{n \geq 4} a^{n-2} \overline{P}_n(\cos E_1) \\
 + Kk^{-3}\mu_{\tilde{E}}(x - x_E) \sum_{n \geq 2} a^{n-2} \overline{P}_n(\cos E_1) &+ Kk^{-3}\mu_{\tilde{S}}(x - x_S) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_S^{n+1}} \overline{P}_n(\cos S_1) \\
 + Kk^{-3}\mu_M(x - x_M) \sum_{n \geq 4} \frac{a^{n-2}}{\tilde{r}_M^{n+1}} \overline{P}_n(\cos M_1) &+ Kk^{-3} \sum_{A \in \mathcal{P}'} \mu_A \left(-\frac{x_{EA}}{\tilde{r}_{EA}^3} + (x - x_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \overline{P}_n(\cos A_1) \right), \\
 \ddot{y} + 2\dot{x} - 3y_E^2y - 3(1 - 2\mu_M)x_E y_E x &= \overline{C(11)}x + \overline{C(12)}y + C(13)z + C(14)\dot{x} + C(15)\dot{y} + C(16)\dot{z} + \overline{C(17)} \\
 + Kk^{-3}[-3(1 - 2m_M)x_E xy - 3y_E y^2] &+ Kk^{-3}(1 - \mu_M)(y - y_E) \sum_{n \geq 4} a^{n-2} \overline{P}_n(\cos E_1) \\
 + Kk^{-3}\mu_{\tilde{E}}(y - y_E) \sum_{n \geq 2} a^{n-2} \overline{P}_n(\cos E_1) &+ Kk^{-3}\mu_{\tilde{S}}(y - y_S) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_S^{n+1}} \overline{P}_n(\cos S_1) \\
 + Kk^{-3}\mu_M(y - y_M) \sum_{n \geq 4} \frac{a^{n-2}}{\tilde{r}_M^{n+1}} \overline{P}_n(\cos M_1) &+ Kk^{-3} \sum_{A \in \mathcal{P}'} \mu_A \left(-\frac{y_{EA}}{\tilde{r}_{EA}^3} + (y - y_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \overline{P}_n(\cos A_1) \right),
 \end{aligned}$$

$$\begin{aligned}
& \ddot{z} + z \\
& = C(21)x + C(22)y + \overline{C(23)}z + C(25)\dot{y} + C(26)\dot{z} + C(27) \\
& + Kk^{-3}(1 - \mu_M)z \sum_{n \geq 3} a^{n-2} \overline{P}_n(\cos E_1) \\
& + Kk^{-3}\mu_E z \sum_{n \geq 2} a^{n-2} \overline{P}_n(\cos E_1) \\
& + Kk^{-3}\mu_S(z - z_S) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_S^{n+1}} \overline{P}_n(\cos S_1) \\
& + Kk^{-3}\mu_M z \sum_{n \geq 3} \frac{a^{n-2}}{\tilde{r}_M^{n+1}} \overline{P}_n(\cos M_1) \\
& + Kk^{-3} \sum_{A \in \mathcal{P}'} \mu_A \left(-\frac{z_{EA}}{\tilde{r}_{EA}^3} + (z - z_A) \sum_{n \geq 2} \frac{a^{n-2}}{\tilde{r}_A^{n+1}} \overline{P}_n(\cos A_1) \right),
\end{aligned}$$

where

$$\begin{aligned}
\mathcal{P}' &= \{S, P_1, \dots, P_k\}, \\
\overline{C(1)} &= C(1) + (Kk^{-3} - 1)(-1 + 3x_E^2), \\
\overline{C(2)} &= C(2) + (Kk^{-3} - 1)3(1 - 2\mu_M)x_E y_E, \\
\overline{C(7)} &= C(7) + Kk^{-3}x_E, \\
\overline{C(11)} &= C(11) + (Kk^{-3} - 1)3(1 - 2\mu_M)x_E y_E, \\
\overline{C(12)} &= C(12) + (Kk^{-3} - 1)(-1 + 3y_E^2), \\
\overline{C(17)} &= C(17) + Kk^{-3}y_E, \\
\overline{C(23)} &= C(23) + (Kk^{-3} - 1)
\end{aligned}$$

and, in fact, \tilde{r}_M is equal to 1 in the normalized system.

The coefficients $C(j)$ and $\overline{C(j)}$ and the ones coming from the effect of the Sun and planets are computed by the routine FUN4. We shall use the following notation for further use:

$$\begin{aligned}
SO(1) &= Kk^{-3} - 1, & SO(6) &= C(5), & SO(11) &= C(16), \\
SO(2) &= \overline{C(1)}, & SO(7) &= \overline{C(7)}, & SO(12) &= \overline{C(17)}, \\
SO(3) &= \overline{C(2)}, & SO(8) &= \overline{C(11)}, & SO(13) &= \overline{C(23)}, \\
SO(4) &= C(3), & SO(9) &= \overline{C(12)}, \\
SO(5) &= C(4), & SO(10) &= C(13)
\end{aligned}$$

for terms and coefficients coming from the noncircular motion of the Moon, and we expand the term

$$Kk^{-1}\mu_S \frac{1}{r_S} \sum_{n \geq 1} \left(\frac{a}{\tilde{r}_S} \right)^n P_n(\cos S_1),$$

accounting for part of the contribution of the Sun to the Hamiltonian as:

$$\text{constant} + \overline{SOS(1)}x + \overline{SOS(2)}y + \overline{SOS(3)}z + SOS(4)x^2 + SOS(5)xy$$

$$+SOS(6)xz + SOS(7)y^2 + SOS(8)yz + SOS(9)z^2 + SOS(10)x^3 + \dots$$

The contributions of the planets can be expanded in a similar way.

Then the contribution of the Sun to the equations is of the following form:

- 1st equation: $SOS(1) + 2SOS(4)x + SOS(5)y + SOS(6)z + \dots$,
- 2nd equation: $SOS(2) + SOS(5)x + 2SOS(7)y + SOS(8)z + \dots$,
- 3rd equation: $SOS(3) + SOS(6)x + SOS(8)y + 2SOS(9)z + \dots$,

where:

$$\begin{aligned} SOS(1) &= \overline{SOS(1)} - Kk^{-3}\mu_S \frac{x_{ES}}{\tilde{r}_{ES}^3}, \\ SOS(2) &= \overline{SOS(2)} - Kk^{-3}\mu_S \frac{y_{ES}}{\tilde{r}_{ES}^3}, \\ SOS(3) &= \overline{SOS(3)} - Kk^{-3}\mu_S \frac{z_{ES}}{\tilde{r}_{ES}^3}, \end{aligned}$$

which is quite useful because of the cancellations (i.e. $\overline{SOS(j)}$, $j = 1, 2, 3$ is quite large but $SOS(j)$, $j = 1, 2, 3$ is rather small.

A Fourier analysis of the functions $SO(i)$, $i = 1, \dots, 13$ and $SOS(i)$, $i = 1, \dots, 9$ has been performed. Other functions $SOS(i)$, $i \geq 10$ have also been computed but the fact that in the denominator appears at least \tilde{r}_S^{-3} makes them negligible.

The Fourier analysis has been done in several steps. First we have performed several FFT in order to identify the relevant frequencies. Some of them appear both in the $SO(i)$ functions and in the $SOS(i)$ ones.

The frequencies retained are in principle, the ones such that the peak of the modulus of the discrete Fourier Transform is greater than 10^{-4} . The unit of frequency has been taken equal to the frequency of the mean longitude of the Moon. Then the frequencies have been checked against linear combinations of the four more important terms appearing in the motion of the Moon (the motion of the Earth–Moon barycenter perihelion has been neglected). Let ν be one of the obtained frequencies. We look for expressions of the type:

$$\nu = n_1\nu_1 + n_2\nu_2 + n_3\nu_3 + n_4\nu_4,$$

where

- ν_1 = frequency of the mean longitude of the Moon = 1,
- ν_2 = frequency of the mean longitude of the lunar perigee,
- ν_3 = frequency of the mean longitude of the ascending node of the Moon,
- ν_4 = frequency of the mean elongation of the Sun.

The FFT have been carried out using 2^{16} points in a time interval equal to 8192 sidereal revolutions of the Moon centered at the epoch $\tau = 1 - 0.5/36525$ Julian centuries since 1900.0. The results obtained for the frequencies are given in the Table C.1 where there appear also the phases at the epoch τ . The values of FFT

are related to the approximate frequencies in the following way:

$$\text{frequency} = \frac{\text{FFT} - 1}{\text{no. revolutions of the secondary}}$$

N	FFT	Frequency	Phase	n_1	n_2	n_3	n_4
1	1088	.1326984719029E + 00	.5820858844883E + 00	2	-2	0	-2
2	6424	.7840427350868E + 00	.3056181124683E + 01	-2	1	0	3
3	6934	.8463731326263E + 00	.2388511629532E + 01	-1	0	1	2
4	7037	.8588467495676E + 00	.1659447913254E + 01	-1	1	0	2
5	7511	.9167412069898E + 00	.3638267009167E + 01	0	-1	0	1
6	7580	.9251959855191E + 00	.5092083509088E + 01	0	0	0	1
7	8124	.9915452214706E + 00	.2241533797741E + 01	1	-1	0	0
8	8226	.1004018838412E + 01	.1512470081464E + 01	1	0	-1	0
9	8328	.1016492455353E + 01	.7834063651862E + 00	1	1	-2	0
10	8736	.1066349235951E + 01	.8448005863156E + 00	2	-1	0	-1
11	14072	.1717693499135E + 01	.3318895826509E + 01	-2	2	0	4
12	14547	.1775587956557E + 01	.5297714922424E + 01	-1	0	0	3
13	15159	.1850391971038E + 01	.3900981710996E + 01	0	0	0	2
14	16246	.1983090442941E + 01	.4483067595483E + 01	2	-2	0	0
15	22195	.2709238720606E + 01	.5560429624249E + 01	-1	1	0	4
16	23282	.2841937192509E + 01	.6142515508738E + 01	1	-1	0	2
17	24369	.2974635664412E + 01	.4414160860469E + 00	3	-3	0	0
18	30318	.3700783942076E + 01	.1518778114812E + 01	0	0	0	4
19	31405	.3833482413979E + 01	.2100863999301E + 01	2	-2	0	2
20	31507	.3845956030921E + 01	.1371800283021E + 01	2	-1	-1	2
21	32594	.3978654502824E + 01	.1953886167510E + 01	4	-3	-1	0

Table C.1. Values of the frequencies and phases.

The normalized time is taken with origin at the epoch τ and such that the sidereal period of the Moon is equal to 2π . Hence if JD is the Julian date counted from 1950.0 (and then $\tau = 18\,262$ JD) we have the relation

$$t_n = (\text{JD} - 18\,262)\bar{n}_M$$

where t_n stands for the normalized time and \bar{n}_M for the mean sidereal motion of the Moon in rad/day (note that $\bar{n}_M = \nu_1$ letting aside the units).

The results of the FFT analysis for the functions $SO(i)$, $i = 1, \dots, 13$ and $SOS(i)$ $i=1, \dots, 9$ have been computed. We list a sample containing $SO(1)$ and $SO(13)$ and all the $SOS(j)$ larger than some tolerance in the Table C.2 and C.3 for the L_4 equilibrium point.

FFT	Cosine coefficient	Sine coefficient
1	$4.736558433505679E - 03$	$0.0E + 00$
1088	$-1.442878806787767E - 03$	$9.486955994339119E - 04$
6424	$1.732884404452373E - 04$	$1.319772662125154E - 03$
7037	$-2.328750090605973E - 03$	$-2.614600756885180E - 02$
7511	$-7.188469510604870E - 04$	$-8.334045780647701E - 04$
7580	$2.527291376289149E - 04$	$6.772719887250550E - 04$
8124	$9.112384873999765E - 02$	$.114730314343880E + 00$
8328	$5.519636493992337E - 04$	$-2.929283508050680E - 04$
8737	$3.427934502822663E - 04$	$4.720421965804989E - 04$
14072	$1.505348196583156E - 04$	$-1.990498598218668E - 05$
14547	$-9.838830990845423E - 04$	$1.041806911006114E - 03$
15159	$-1.434663778272931E - 02$	$1.362093368173893E - 02$
16246	$2.038370016127341E - 03$	$-8.718756403069540E - 03$
22195	$5.837752807074912E - 04$	$5.138938756818489E - 04$
23282	$-3.977169383514819E - 03$	$-5.612676063410181E - 04$
24369	$9.015594811462270E - 04$	$-4.245134436628341E - 04$
30318	$-4.837067869558339E - 06$	$1.571617391046836E - 04$
31405	$-1.194870945858466E - 04$	$-2.090091434778585E - 04$
1	$-6.346799886093270E - 03$	$0.0E + 00$
1088	$1.871401381441822E - 03$	$-1.229933345372316E - 03$
6424	$-2.093906569989454E - 04$	$-1.610294549322988E - 03$
7037	$2.926146024377030E - 03$	$3.282871340455004E - 02$
7511	$9.223549824103249E - 04$	$1.067890991980766E - 03$
7580	$-3.222400272087621E - 04$	$-8.689878798638993E - 04$
7649	$7.062830239169652E - 05$	$1.117174376136566E - 04$
8124	$-.120996727832802E + 00$	$-.152343904826198E + 00$
8328	$-7.390184267466767E - 04$	$3.950008491334309E - 04$
8737	$-4.699097976667718E - 04$	$-6.493835058819712E - 04$
9211	$1.078825676320622E - 04$	$1.036011623173255E - 05$
14547	$1.939818528408744E - 03$	$-2.048900191259839E - 03$
15159	$2.887777795874210E - 02$	$-2.741286236759826E - 02$
16246	$-3.150038221555223E - 03$	$1.347080813131899E - 02$
22195	$-1.235076277776033E - 03$	$-1.088438002013584E - 03$
23282	$8.348499106275216E - 03$	$1.178953359826934E - 03$
24369	$-1.551432331495733E - 03$	$7.315205923789502E - 04$
30318	$-1.991186202809775E - 05$	$2.910381907009060E - 04$
31405	$-1.924523895413760E - 04$	$-3.200122329176353E - 04$

Table C.2. FFT of functions $SO(1)$ (top): number of terms = 18 and $SO(13)$ (bottom): number of terms = 19.

FFT	Cosine coefficient	Sine coefficient
1	$1.076041971379874E-03$	$0.0E+00$
7037	$5.540968232159205E-04$	$2.576994925933965E-04$
8124	$-1.371567874626327E-04$	$2.223408884711102E-05$
14547	$9.452870104655420E-05$	$3.175589405929647E-04$
15159	$1.146265957232343E-03$	$4.775571350912270E-03$
22195	$1.297644728743069E-04$	$-4.380310979254232E-05$
23282	$-4.299773470473283E-04$	$5.490768205089372E-04$

Table C.3. FFT of function $SOS(1)$. Total number of terms = 7. Factor = 0.8.

FFT	Cosine coefficient	Sine coefficient
1	$9.849081716565741E-04$	$0.0E+00$
7037	$1.291139808833131E-04$	$-2.767737044324544E-04$
14547	$1.587833276134036E-04$	$-4.736270317474194E-05$
15159	$2.387847913536772E-03$	$-5.735836085890310E-04$
23282	$2.744346740034203E-04$	$2.148819384748705E-04$

Table C.3 (Continuation). FFT of function $SOS(2)$. Total number of terms = 5.
Factor = 0.4.

FFT	Cosine coefficient	Sine coefficient
1	$8.608527472909735E-04$	$0.0E+00$
7037	$2.137106439286387E-05$	$2.433446750792411E-04$
14547	$-9.112096775198372E-05$	$9.635116100675502E-05$
15159	$-1.425152968611055E-03$	$1.352586742233675E-03$
23282	$-2.761985649308407E-04$	$-3.906650547346975E-05$

Table C.3 (Continuation). FFT of function $SOS(4)$. Total number of terms = 5.
Factor = 0.64.

FFT	Cosine coefficient	Sine coefficient
7037	$2.435908829381741E-04$	$-2.147050054569744E-05$
14547	$9.627416669766357E-05$	$9.104857039991896E-05$
15159	$1.352265753433075E-03$	$1.424840670512157E-03$
23282	$-3.913691359113510E-05$	$2.761690038753613E-04$

Table C.3 (Continuation). FFT of function $SOS(5)$. Total number of terms = 4.
Factor = 0.32.

FFT	Cosine coefficient	Sine coefficient
1	2.274585745896600E - 04	0.0E + 00
15159	3.562730984701729E - 04	-3.381233036219169E - 04

Table C.3 (End). FFT of function $SOS(7)$. Total number of terms = 2. Factor = 0.16.

The output of the FFT as it is given has been filtered using the following criteria: Only the modulus of an harmonic is considered. If this modulus times a certain factor is greater than a certain tolerance (10^{-4}) then the term is retained. The factor has been taken equal to one for all the $SO(i)$ functions and it is explicitly given for the $SOS(i)$ functions in the tables. Its value depends on the power of x , y , z which multiplies the function in the equations.

When the frequencies are known it is possible to obtain the related coefficients by means of the Fourier integral

$$\frac{1}{T} \int_{t_0}^{t_0+T} e^{-it} f(t) dt,$$

where the time interval T is a multiple of the period associated to the frequency. The integral has been discretized using the trapezoidal rule. Several time intervals and steps have been used in the numerical integration to allow for interval checks. This task is performed by routine FURLE4 of program FURI4. The routine analyzes, for a given frequency all the functions. In the Tables C.4, C.5 and C.6 we present a sample of the results obtained by this routine for the first three frequencies.

SO(1)	.17382394166E-02	-.59997788432E-05	.47395438954E-02
SO(2)	.10254938500E-02	-.44121424690E-05	.36027986003E-02
SO(3)	.22031065393E-02	.38134680368E-04	.60072135347E-02
SO(4)	-.17639764051E-06	.19321594920E-06	-.17353689874E-07
SO(5)	-.17307793543E-05	.72200869768E-04	-.90353795223E-06
SO(6)	-.26882174545E-03	-.53853805348E-05	.32977405368E-04
SO(7)	-.13914423698E-03	.39655171023E-04	.24057388853E-04
SO(8)	.22032251981E-02	-.53343756680E-04	.60072434427E-02
SO(9)	.36333639998E-02	-.13413292051E-04	.10713307661E-01
SO(10)	.10485703027E-05	-.11465982572E-06	.41263815965E-06
SO(11)	-.92204252159E-07	-.12856628128E-06	-.90175501289E-07
SO(12)	-.24044366876E-03	-.22794948912E-04	.42731885096E-04
SO(13)	-.22537364397E-02	.78832163123E-05	-.63511715714E-02

Table C.4. Fourier coefficients of the $SO(i)$ and $SOS(i)$ functions, related to the frequency number 1. The first column corresponds to the cosine coefficients, the second one to the sine ones, and the last one to the independent term. The integration has been done for 8192 revolutions of the secondary and taking 8 points in each revolution.

SO(1)	.30632775366E-03	.12346476348E-02	.47341287409E-02
SO(2)	.24249374638E-03	.96888738711E-03	.36012017019E-02
SO(3)	.33831721635E-03	.15780928718E-02	.60005784389E-02
SO(4)	-.13503954382E-05	.14476202672E-05	.42641555370E-06
SO(5)	.57330463257E-03	-.14472472257E-03	-.49781903353E-05
SO(6)	.35050838445E-03	.14174950728E-02	.29956955699E-04
SO(7)	-.36878758031E-04	.32894793201E-04	.25487464934E-04
SO(8)	.43820510280E-03	.15516641720E-02	.60001514495E-02
SO(9)	.70196530135E-03	.28207723232E-02	.10703587254E-01
SO(10)	-.13869493463E-05	-.10175523278E-05	-.55271606980E-06
SO(11)	.28201432971E-05	-.27481969897E-05	-.17430992628E-05
SO(12)	.35994617897E-04	.30471828167E-04	.44751279129E-04
SO(13)	-.36918032250E-03	-.15037606804E-02	-.63420936877E-02

Table C.5. Same as Table C.4 but for the frequency number 2. Number of revolutions: 4096. Points per revolution: 16.

SO(1)	-.12594557653E-02	-.10753826679E-02	.46883313243E-02
SO(2)	-.97557845230E-03	-.83490712227E-03	.35655706288E-02
SO(3)	-.15752246597E-02	-.13900301966E-02	.59405518526E-02
SO(4)	.20009741744E-05	-.72477983103E-03	-.11795161652E-05
SO(5)	-.63634153151E-03	.67359537336E-03	-.32478296958E-04
SO(6)	-.15605501736E-02	-.13249168303E-02	-.38796341658E-04
SO(7)	.27775849432E-05	-.37580911560E-04	.23131449008E-04
SO(8)	-.16174194937E-02	-.13359997934E-02	.59440843559E-02
SO(9)	-.28647541603E-02	-.24479837772E-02	.10599262504E-01
SO(10)	.60705362417E-03	.17137791843E-05	.43077937962E-06
SO(11)	-.40866601041E-05	.14431431247E-02	.51020775047E-05
SO(12)	-.37377039494E-04	-.11063942554E-04	.44632152297E-04
SO(13)	.16058308817E-02	.13682229965E-02	-.62825187859E-02
SOS(1)	.34000091963E-04	-.10493905037E-04	.13450566027E-02
SOS(2)	-.10881522867E-04	-.34846744803E-04	.24626744913E-02
SOS(3)	.63677501510E-03	.36331197235E-03	-.33152319641E-06
SOS(4)	.13120464456E-04	.12482105204E-04	.13455711899E-02
SOS(5)	.24023399950E-04	-.26527334802E-04	-.57890491722E-06
SOS(6)	-.91503256635E-06	.72517896787E-03	.11565824739E-05
SOS(7)	-.13263767627E-04	-.12414720462E-04	.14220109638E-02
SOS(8)	.73582861048E-03	.84265032711E-06	-.10549880246E-05
SOS(9)	.14330317099E-06	-.67384742087E-07	-.27675821537E-02

Table C.6. Same as Table C.4 but for the frequency number 3. Number of revolutions: 2048. Points per revolution: 32.

With these values in hand we have performed a second Fourier analysis of the real functions minus the values obtained by the Fourier computations (routine FUN4A computes these new modified functions). This second step can be seen as a check of the results obtained and also allows the detection of other frequencies, very close to the previously detected and having a not too small amplitude. Though there appear some new frequencies it has been seen that their contribution is not significant. A sample of the results obtained by routine FURLEN4 for these modified functions is given in the Tables C.7, C.8 and C.9 for the first three frequencies. It can be seen that the coefficients obtained for the cosine and sine terms are rather small.

$SO(1)$	$.35568417572E - 06$	$-.40322419498E - 06$	$.28559174786E - 06$
$SO(2)$	$.70115792837E - 06$	$.38139633818E - 06$	$-.29681465409E - 06$
$SO(3)$	$.12146877298E - 07$	$.48569426731E - 06$	$.64051007480E - 06$
$SO(4)$	$-.15731356753E - 06$	$.23708440564E - 06$	$.16329791151E - 08$
$SO(5)$	$-.21103029284E - 06$	$.75922810896E - 06$	$.23284808734E - 06$
$SO(6)$	$-.38749253552E - 06$	$-.97659341736E - 07$	$.14632035060E - 06$
$SO(7)$	$.32141020623E - 07$	$.48594035631E - 06$	$.39486426620E - 09$
$SO(8)$	$-.20011280866E - 06$	$-.49103131773E - 06$	$.62693609842E - 06$
$SO(9)$	$.24369851555E - 06$	$-.25774766661E - 06$	$-.89674748628E - 04$
$SO(10)$	$.13989979809E - 06$	$.11957476295E - 06$	$-.18440157474E - 07$
$SO(11)$	$.41108634080E - 06$	$-.60856366639E - 06$	$.17516324602E - 08$
$SO(12)$	$-.12746328656E - 05$	$-.26368164664E - 07$	$-.41002471794E - 06$
$SO(13)$	$-.73948979187E - 07$	$.35524554515E - 06$	$-.38449423876E - 06$

Table C.7. Fourier coefficients of the modified $SO(i)$ and $SOS(i)$ functions, related to the frequency number 1. The integration has been done as in Table C.4.

$SO(1)$	$-.31467465776E - 06$	$.70569124026E - 07$	$.69848479382E - 06$
$SO(2)$	$.23324209725E - 06$	$.41624917891E - 06$	$-.29517876782E - 06$
$SO(3)$	$.33006088003E - 06$	$-.56178918144E - 07$	$.11569840516E - 05$
$SO(4)$	$-.52035238471E - 07$	$.67768182828E - 07$	$-.71518814952E - 07$
$SO(5)$	$-.40097876687E - 06$	$-.32471236087E - 06$	$.32022021226E - 08$
$SO(6)$	$-.22078701162E - 06$	$.45483402084E - 06$	$.31237004204E - 06$
$SO(7)$	$-.19852744985E - 06$	$.30875099175E - 08$	$-.16030402325E - 06$
$SO(8)$	$-.39933817738E - 07$	$-.18197969459E - 06$	$.11573071351E - 05$
$SO(9)$	$.24188191871E - 06$	$-.59286057368E - 07$	$-.89053298322E - 04$
$SO(10)$	$-.61594687680E - 07$	$-.41949297687E - 07$	$.63634463143E - 07$
$SO(11)$	$.80007061269E - 07$	$-.15246825854E - 06$	$.27512674965E - 06$
$SO(12)$	$.19118334596E - 06$	$.41542217047E - 06$	$-.65647123209E - 06$
$SO(13)$	$.46797557741E - 06$	$-.44283782648E - 06$	$-.94246025229E - 06$

Table C.8. Same as Table C.7 but for the frequency number 2. The integration as in Table C.5.

$SO(1)$	$.83862202057E - 07$	$.12053975395E - 06$	$.75000519757E - 06$
$SO(2)$	$.15084988671E - 06$	$.21198080749E - 06$	$.41922643094E - 07$
$SO(3)$	$.10131300913E - 06$	$.78747132178E - 07$	$.12087952354E - 05$
$SO(4)$	$-.14977378829E - 07$	$-.45068790493E - 06$	$.17041202246E - 06$
$SO(5)$	$.45668273592E - 07$	$-.42364167443E - 07$	$.77235136276E - 08$
$SO(6)$	$.21389373857E - 06$	$.16964795833E - 06$	$.61346136009E - 06$
$SO(7)$	$.47234336885E - 07$	$-.11750549388E - 07$	$-.19035975416E - 07$
$SO(8)$	$.11483834669E - 06$	$.23676579824E - 06$	$.12361078470E - 05$
$SO(9)$	$.28735451872E - 06$	$.14052713876E - 05$	$-.88639938983E - 04$
$SO(10)$	$.74150799164E - 06$	$.25949499804E - 08$	$-.11900690689E - 06$
$SO(11)$	$.86156201155E - 07$	$.36440373236E - 06$	$-.67459315737E - 06$
$SO(12)$	$.58220691088E - 05$	$.16316368674E - 05$	$-.32477746550E - 06$
$SO(13)$	$-.98281403861E - 07$	$-.16817065629E - 06$	$-.10120677298E - 05$

Table C.9. Same as Table C.7 but for the frequency number 3. The integration as in Table C.6.

C.7 Simplified Normalized Equations

We shall use a simplified version of the equations of motion that retains the terms called $SO(i)$ and $SOS(i)$ and the ones coming from the RTBP.

Let us introduce some auxiliary functions $P(i)$, $i=1,...,20$ which account for the terms appearing in $SO(i)$ and $SOS(i)$ according to the following definition

$$\begin{aligned}
P(1) &= SOS(1) + SO(7), \\
P(2) &= 1 + 2SOS(4) + 0.25SO(1) + \alpha SO(2), \\
P(3) &= SOS(5) + SO(3) + \alpha SO(1), \\
P(4) &= SOS(6) + SO(4), \\
P(5) &= SO(5), \\
P(6) &= 2 + SO(6), \\
P(7) &= 1 + SO(1), \\
P(8) &= SOS(2) + SO(12), \\
P(9) &= SOS(5) + SO(8) + \alpha SO(1), \\
P(10) &= 1 + 2SOS(7) + SO(9) - 1.25SO(1), \\
P(11) &= SOS(8) + SO(10), \\
P(12) &= -2 - SO(6), \\
P(13) &= SO(5), \\
P(14) &= SO(11), \\
P(15) &= SOS(3) + y_E SO(10) + 0.5SO(4), \\
P(16) &= SOS(6) + SO(4), \\
P(17) &= SOS(8) - SO(10), \\
P(18) &= 2SOS(9) + SO(13) + SO(1), \\
P(19) &= -SO(11),
\end{aligned}$$

$$P(20) = SO(5),$$

where α stands for $1.5y_E(1 - 2\mu_M)$. With these definitions the functions $P(i)$ can be expressed as

$$P(i) = A_{i,0} + \sum_{j=1}^m A_{i,j} \cos \theta_j + \sum_{j=1}^m B_{i,j} \sin \theta_j,$$

where $\theta_j = \nu_j t + \varphi_j$, with ν_j , φ_j being the frequency and phase as given in the preceding section, and t denotes the normalized time. The values of the coefficients, given in a suitable way to be used by the program that computes quasi-periodic orbits, are given in Table C.10 .

1						.00137	2	0	0	0	4	0	.00102
1	2	-2	0	-2	0	-.00014	2	2	-2	0	2	0	.00109
1	-1	1	0	2	1	.00158	2	4	-3	-1	0	0	.00006
1	1	-1	0	0	0	.00031	3						.00000
1	1	-1	0	0	1	.00017	3	-2	1	0	3	0	-.00005
1	-2	2	0	4	0	.00033	3	-1	1	0	2	1	.00181
1	-2	2	0	4	1	-.00031	3	1	-1	0	0	1	.00020
1	-1	0	0	3	0	.00086	3	-2	2	0	4	1	-.00036
1	-1	0	0	3	1	-.00020	3	-1	0	0	3	0	.00100
1	0	0	0	2	1	-.01428	3	-1	0	0	3	1	-.00021
1	-1	1	0	4	0	-.00011	3	0	0	0	2	1	-.01649
1	-1	1	0	4	1	-.00015	3	-1	1	0	4	1	-.00017
1	1	-1	0	2	1	-.00154	3	1	-1	0	2	1	-.00179
1	0	0	0	4	0	.00042	3	0	0	0	4	1	.00014
1	0	0	0	4	1	.00011	3	2	-2	0	2	0	-.00006
1	2	-2	0	2	0	.00037	3	2	-2	0	2	1	.00032
1	2	-2	0	2	1	.00026	3	2	-1	-1	2	0	.00009
1	2	-1	-1	2	0	.00015	3	4	-3	-1	0	1	-.00006
1	2	-1	-1	2	1	-.00005	4						.00000
2						1.00759	4	2	-1	-1	2	1	-.00102
2	2	-2	0	-2	0	.00145	4	4	-3	-1	0	1	-.00029
2	2	-2	0	-2	1	-.00010	5						.00000
2	-2	1	0	3	0	.00031	5	2	-2	0	-2	1	.00007
2	-2	1	0	3	1	.00136	5	-2	1	0	3	0	.00062
2	-1	1	0	2	0	.03148	5	-2	1	0	3	1	-.00014
2	0	-1	0	1	0	.00023	5	-1	1	0	2	1	-.01686
2	0	-1	0	1	1	.00098	5	0	-1	0	1	0	.00060
2	0	0	0	1	0	-.00089	5	0	-1	0	1	1	-.00014
2	1	-1	0	0	0	.16503	5	0	0	0	1	1	.00053
2	1	1	-2	0	0	-.00062	5	1	-1	0	0	1	-.10793
2	2	-1	0	-1	0	-.00018	5	1	1	-2	0	1	.00043
2	2	-1	0	-1	1	.00081	5	2	-1	0	-1	0	.00058
2	-2	2	0	4	0	.00084	5	2	-1	0	-1	1	.00013
2	-1	0	0	3	0	.00028	5	-2	2	0	4	1	.00010
2	-1	0	0	3	1	.00122	5	-1	0	0	3	0	.00190
2	0	0	0	2	0	.02665	5	-1	0	0	3	1	-.00044
2	2	-2	0	0	0	.01351	5	0	0	0	2	1	-.02947
2	-1	1	0	4	0	.00057	5	2	-2	0	0	1	-.00882
2	1	-1	0	2	0	.00423	5	-1	1	0	4	1	-.00074
2	3	-3	0	0	0	.00107	5	1	-1	0	2	1	-.00382

5	3	-3	0	0	1	-.00068	7	2	-2	0	0	0	.01344
5	0	0	0	4	1	.00017	7	-1	1	0	4	0	.00079
5	2	-2	0	2	1	.00024	7	1	-1	0	2	0	.00417
6						2.00003	7	3	-3	0	0	0	.00107
6	2	-2	0	-2	0	-.00027	7	0	0	0	4	0	.00016
6	-2	1	0	3	0	.00035	7	2	-2	0	2	0	.00025
6	-2	1	0	3	1	.00152	8						.00251
6	-1	1	0	2	0	.03817	8	2	-2	0	-2	0	-.00024
6	0	-1	0	1	0	.00029	8	-1	1	0	2	0	.00165
6	0	-1	0	1	1	.00127	8	1	-1	0	0	0	.00054
6	0	0	0	1	0	-.00112	8	1	-1	0	0	1	-.00010
6	1	-1	0	0	0	.21765	8	-2	2	0	4	0	.00057
6	1	1	-2	0	0	-.00083	8	-2	2	0	4	1	.00017
6	2	-1	0	-1	0	-.00025	8	-1	0	0	3	0	-.00020
6	2	-1	0	-1	1	.00110	8	-1	0	0	3	1	-.00090
6	-2	2	0	4	0	.00051	8	0	0	0	2	0	-.01435
6	-1	0	0	3	0	.00064	8	2	-2	0	0	0	.00006
6	-1	0	0	3	1	.00279	8	-1	1	0	4	0	-.00018
6	0	0	0	2	0	.04294	8	-1	1	0	4	1	.00008
6	2	-2	0	0	0	.01481	8	1	-1	0	2	0	-.00161
6	-1	1	0	4	0	.00101	8	0	0	0	4	0	.00046
6	1	-1	0	2	0	.00535	8	0	0	0	4	1	-.00023
6	3	-3	0	0	0	.00104	8	2	-2	0	2	0	.00076
7						1.00474	8	2	-2	0	2	1	-.00019
7	2	-2	0	-2	0	.00173	8	2	-1	-1	2	1	.00005
7	2	-2	0	-2	1	-.00011	9						.00000
7	-2	1	0	3	0	.00030	9	-2	1	0	3	0	.00005
7	-2	1	0	3	1	.00132	9	1	-1	0	0	1	-.00020
7	-1	1	0	2	0	.03149	9	-2	2	0	4	1	.00036
7	0	-1	0	1	0	.00023	9	-1	1	0	4	1	.00017
7	0	-1	0	1	1	.00100	9	0	0	0	4	1	-.00034
7	0	0	0	1	0	-.00086	9	2	-2	0	2	0	.00006
7	1	-1	0	0	0	.16441	9	2	-2	0	2	1	-.00032
7	1	1	-2	0	0	-.00063	9	2	-1	-1	2	0	-.00009
7	2	-1	0	-1	0	-.00018	9	4	-3	-1	0	1	.00005
7	2	-1	0	-1	1	.00080	10						1.00768
7	-2	2	0	4	0	.00019	10	2	-2	0	-2	0	.00145
7	-1	0	0	3	0	.00040	10	2	-2	0	-2	1	-.00010
7	-1	0	0	3	1	.00172	10	-2	1	0	3	0	.00031
7	0	0	0	2	0	.02658	10	-2	1	0	3	1	.00136

10	-1	1	0	2	0	.03348	12	-1	0	0	3	0	-.00064
10	0	-1	0	1	0	.00023	12	-1	0	0	3	1	-.00279
10	0	-1	0	1	1	.00099	12	0	0	0	2	0	-.04294
10	0	0	0	1	0	-.00088	12	2	-2	0	0	0	-.01481
10	1	-1	0	0	0	.16504	12	-1	1	0	4	0	-.00101
10	1	1	-2	0	0	-.00062	12	1	-1	0	2	0	-.00535
10	2	-1	0	-1	0	-.00019	12	3	-3	0	0	0	-.00104
10	2	-1	0	-1	1	.00080	13						.00000
10	-2	2	0	4	0	.00084	13	2	-2	0	-2	1	.00007
10	-1	0	0	3	0	.00028	13	-2	1	0	3	0	.00062
10	-1	0	0	3	1	.00121	13	-2	1	0	3	1	-.00014
10	0	0	0	2	0	.01014	13	-1	1	0	2	1	-.01686
10	2	-2	0	0	0	.01351	13	0	-1	0	1	0	.00060
10	-1	1	0	4	0	.00057	13	0	-1	0	1	1	-.00014
10	1	-1	0	2	0	.00223	13	0	0	0	1	1	.00053
10	3	-3	0	0	0	.00107	13	1	-1	0	0	1	-.10793
10	0	0	0	4	0	.00063	13	1	1	-2	0	1	.00043
10	2	-2	0	2	0	.00109	13	2	-1	0	-1	0	.00058
10	2	-1	-1	2	1	.00007	13	2	-1	0	-1	1	.00013
10	4	-3	-1	0	0	.00007	13	-2	2	0	4	1	.00010
11						.00000	13	-1	0	0	3	0	.00190
11	-1	0	1	2	0	.00131	13	-1	0	0	3	1	-.00044
11	1	0	-1	0	0	-.00142	13	0	0	0	2	1	-.02947
11	0	0	0	4	0	.00008	13	2	-2	0	0	1	-.00882
11	0	0	0	4	1	.00006	13	-1	1	0	4	1	-.00074
11	2	-1	-1	2	0	.00391	13	1	-1	0	2	1	-.00382
11	4	-3	-1	0	0	.00116	13	3	-3	0	0	1	-.00068
12						-2.00003	13	0	0	0	4	1	.00017
12	2	-2	0	-2	0	.00027	13	2	-2	0	2	1	.00024
12	-2	1	0	3	0	-.00035	14						.00000
12	-2	1	0	3	1	-.00152	14	-1	0	1	2	1	.00144
12	-1	1	0	2	0	-.03817	14	1	0	-1	0	1	-.00145
12	0	-1	0	1	0	-.00029	14	2	-1	-1	2	1	.00203
12	0	-1	0	1	1	-.00127	14	4	-3	-1	0	1	.00058
12	0	0	0	1	0	.00112	15						.00000
12	1	-1	0	0	0	-.21765	15	-1	0	1	2	0	.00007
12	1	1	-2	0	0	.00083	15	0	0	0	4	0	-.00006
12	2	-1	0	-1	0	.00025	15	0	0	0	4	1	-.00006
12	2	-1	0	-1	1	-.00110	15	2	-1	-1	2	0	-.00339
12	-2	2	0	4	0	-.00051	15	2	-1	-1	2	1	-.00050

15	4	-3	-1	0	0	-.00101	18	0	0	0	4	0	.00046
15	4	-3	-1	0	1	-.00018	18	2	-2	0	2	0	.00063
16						.00000	18	2	-1	-1	2	0	.00008
16	2	-1	-1	2	1	-.00102	18	2	-1	-1	2	1	-.00008
16	4	-3	-1	0	1	-.00029	19						.00000
17						.00000	19	-1	0	1	2	1	-.00144
17	-1	0	1	2	0	.00009	19	1	0	-1	0	1	.00145
17	0	0	0	4	0	-.00008	19	2	-1	-1	2	1	-.00203
17	0	0	0	4	1	-.00006	19	4	-3	-1	0	1	-.00058
17	2	-1	-1	2	0	-.00391	20						.00000
17	4	-3	-1	0	0	-.00116	20	2	-2	0	-2	1	.00007
18						-.00721	20	-2	1	0	3	0	.00062
18	2	-2	0	-2	0	-.00051	20	-2	1	0	3	1	-.00014
18	-2	1	0	3	0	-.00007	20	-1	1	0	2	1	-.01686
18	-2	1	0	3	1	-.00029	20	0	-1	0	1	0	.00060
18	-1	1	0	2	0	-.00805	20	0	-1	0	1	1	-.00014
18	0	-1	0	1	0	-.00006	20	0	0	0	1	1	.00053
18	0	-1	0	1	1	-.00028	20	1	-1	0	0	1	-.10793
18	0	0	0	1	0	.00024	20	1	1	-2	0	1	.00043
18	1	-1	0	0	0	-.05390	20	2	-1	0	-1	0	.00058
18	1	1	-2	0	0	.00022	20	2	-1	0	-1	1	.00013
18	2	-1	0	-1	0	.00007	20	-2	2	0	4	1	.00010
18	2	-1	0	-1	1	-.00030	20	-1	0	0	3	0	.00190
18	-2	2	0	4	0	.00010	20	-1	0	0	3	1	-.00044
18	-1	0	0	3	0	-.00038	20	0	0	0	2	1	-.02947
18	-1	0	0	3	1	-.00166	20	2	-2	0	0	1	-.00882
18	0	0	0	2	0	-.02691	20	-1	1	0	4	1	-.00074
18	2	-2	0	0	0	-.00733	20	1	-1	0	2	1	-.00382
18	-1	1	0	4	0	-.00088	20	3	-3	0	0	1	-.00068
18	1	-1	0	2	0	-.00459	20	0	0	0	4	1	.00017
18	3	-3	0	0	0	-.00078	20	2	-2	0	2	1	.00024

Table C.10. Values of $A_{i,j}$ and $B_{i,j}$. The first index gives the number, i , of the function $P(i)$. The four next indices denote the linear combination of the four basic frequencies. The fifth one is 0 for the $A_{i,j}$ coefficients (cosine terms) and 1 for the $B_{i,j}$ ones (sine terms). Only coefficients greater than 10^{-5} have been retained.

With this notation the equations of motion written in a simplified form are:

$$\begin{aligned}
 \ddot{x} &= P(7) \left[-\frac{x - x_E}{r_{PE}^3} (1 - \mu_M) - \frac{x + x_E}{r_{PM}^3} \mu_M - x_E (1 - 2\mu_M) \right] \\
 &\quad + P(1) + P(2)x + P(3)y + P(4)z + P(5)\dot{x} + P(6)\dot{y}, \\
 \ddot{y} &= P(7) \left[-\frac{y - y_E}{r_{PE}^3} (1 - \mu_M) - \frac{y - y_E}{r_{PM}^3} \mu_M - y_E \right] + P(8) + P(9)x \\
 &\quad + P(10)y + P(11)z + P(12)\dot{x} + P(13)\dot{y} + P(14)\dot{z}, \\
 \ddot{z} &= P(7) \left[-\frac{z}{r_{PE}^3} (1 - \mu_M) - \frac{z}{r_{PM}^3} \mu_M \right] + P(15) + P(16)x + P(17)y
 \end{aligned}$$

$$+P(18)z + P(19)\dot{y} + P(20)\dot{z},$$

where r_{PE} , r_{PM} denote the distances from the particle to the Earth and Moon, respectively, given by

$$r_{PE}^2 = (x - x_E)^2 + (y - y_E)^2 + z^2,$$

$$r_{PM}^2 = (x + x_E)^2 + (y - y_E)^2 + z^2.$$

We recall that $x_E = -1/2$, $y_E = -\sqrt{3}/2$ for L_4 and $x_E = -1/2$, $y_E = \sqrt{3}/2$ for L_5 .

C.8 Numerical Tests

To be sure that, the simplified system is meaningful we have written a test program (FUCHE4) such that given initial conditions in normalized coordinates, transforms them to ecliptic ones. Then computes the vector field on the particle in ecliptic coordinates using the full solar system (JPL ephemeris) and after performs the transformation back to normalized coordinates. On the other hand we can evaluate directly the simplified vector field (already in normalized coordinates). Then we can select either a given point in the neighborhood of L_4 in the phase space (introducing the position and velocity) or some points from a quasi-periodic orbit (we have used the planar one mentioned in 1.5.1)

The biggest difference between the two vector fields in normalized coordinates, is of the order of 10^{-3} . This is due to the fact that we are using a simplified version of the equations in which the time-dependent coefficients are functions of an small set of frequencies (21). In the other hand the amplitudes related to these frequencies have been computed using an analytic model of the solar system and here the tests have been done against a numerical one. In any case it seems reasonable to believe that the orbits found using the simplified equations can be easily modified to satisfy the complete system of equations. Below we give some results of these tests.

First we give the modified Julian day (origin at 1950.0). Then the initial position and velocity in normalized units. The vector fields computed by Newton's law using the JPL ephemeris for the full solar system, and by the simplified model are given. Finally the differences are printed.

Day =18000.0

	Position			Velocity	
.4331E-02	.4603E-02	.0000E+00	.7472E-02	.4518E-01	.0000E+00
	JPL			Fourier	
.121E+00	-9.548E-04	8.522E-04	.123E+00	2.325E-03	4.896E-04
	Differences				
	2.190E-03	3.280E-03	-3.625E-04		

Day =18004.0

	Position			Velocity	
.2308E-01	-.9956E-02	.0000E+00	.4378E-02	-.5727E-01	.0000E+00
	JPL			Fourier	
-.134E+00	-7.246E-03	-1.548E-03	-.133E+00	-5.169E-03	-2.223E-03
	Differences				
	1.309E-03	2.076E-03	-6.751E-04		

Day =18008.0

	Position			Velocity	
.2546E-01	-.6202E-01	.0000E+00	.8812E-02	-.3001E-01	.0000E+00
	JPL			Fourier	
-.140E+00	-.113E+00	2.610E-03	-.138E+00	-.113E+00	3.198E-03
	Differences				
	2.155E-03	7.484E-04	5.880E-04		

Day =18012.0

	Position			Velocity	
.1285E-01	-.2503E-01	.0000E+00	-.4244E-01	.9153E-01	.0000E+00
	JPL			Fourier	
.170E+00	3.792E-02	-3.931E-03	.171E+00	4.056E-02	-3.941E-03
	Differences				
	1.216E-03	2.634E-03	-9.854E-06		

Day =18016.0

	Position			Velocity		
.1858E-01	.5904E-02	.0000E+00	.2784E-01	-.2869E-01	.0000E+00	
	JPL			Fourier		
-3.038E-02	-3.016E-02	3.893E-03	-2.978E-02	-2.862E-02	4.030E-03	
Differences						
	5.976E-04	1.547E-03	1.364E-04			

Day =18020.0

	Position			Velocity		
.2124E-01	-.4192E-01	.0000E+00	.3842E-02	-.6128E-01	.0000E+00	
	JPL			Fourier		
-.157E+00	-7.021E-02	-2.786E-03	-.156E+00	-6.807E-02	-3.148E-03	
Differences						
	5.854E-04	2.135E-03	-3.624E-04			

Day =18024.0

	Position			Velocity		
.2540E-01	-.5558E-01	.0000E+00	-.2128E-01	.5774E-01	.0000E+00	
	JPL			Fourier		
5.146E-02	-2.504E-02	1.852E-03	5.331E-02	-2.191E-02	2.032E-03	
Differences						
	1.848E-03	3.132E-03	1.802E-04			

Day =18028.0

	Position			Velocity		
.7192E-02	.7655E-02	.0000E+00	.2130E-01	.2310E-01	.0000E+00	
	JPL			Fourier		
8.352E-02	-1.359E-02	-1.498E-04	8.516E-02	-1.070E-02	-1.413E-04	
Differences						
	1.638E-03	2.886E-03	8.434E-06			

Day =18032.0

Position			Velocity		
.2201E-01	-.1775E-01	.0000E+00	-.4279E-02	-.5752E-01	.0000E+00
JPL			Fourier		
-.141E+00	-1.437E-02	-1.393E-03	-.140E+00	-1.164E-02	-1.664E-03
Differences					
	1.151E-03	2.727E-03		-2.703E-04	

Day =18140.0

Position			Velocity		
.1994E-01	-.3135E-02	.0000E+00	.7127E-01	-.3114E-01	.0000E+00
JPL			Fourier		
-6.424E-02	1.317E-02	-1.269E-03	-6.223E-02	1.616E-02	-2.325E-03
Differences					
	2.012E-03	2.983E-03		-1.055E-03	

Day =18144.0

Position			Velocity		
.7461E-02	-.3875E-01	.0000E+00	-.1026E-01	-.3574E-01	.0000E+00
JPL			Fourier		
-.116E+00	-5.530E-02	6.615E-04	-.115E+00	-5.254E-02	1.920E-03
Differences					
	1.311E-03	2.752E-03		1.259E-03	

Day =18148.0

Position			Velocity		
.1485E-01	-.4205E-01	.0000E+00	.3903E-02	.4640E-01	.0000E+00
JPL			Fourier		
5.352E-02	-.101E+00	-2.899E-04	5.459E-02	-9.964E-02	-1.476E-03
Differences					
	1.064E-03	1.606E-03		-1.186E-03	

Day =18152.0

Position		Velocity	
.1155E-01	.3235E-02	.0000E+00	.2604E-01
.1447E-03	.0000E+00		
JPL		Fourier	
-4.164E-03	-3.256E-02	9.665E-04	-2.599E-03
-3.118E-02			
Differences			
1.564E-03	1.386E-03	2.555E-05	

Day =18156.0

Position		Velocity	
.1307E-01	-.2273E-01	.0000E+00	-.2854E-01
-.3552E-01	.0000E+00		
JPL		Fourier	
-8.436E-02	3.492E-02	-4.509E-04	-8.362E-02
3.737E-02			
Differences			
7.454E-04	2.450E-03	4.503E-04	

Day =18160.0

Position		Velocity	
.9541E-02	-.4964E-01	.0000E+00	.1933E-01
-.3819E-02	.0000E+00		
JPL		Fourier	
-4.872E-02	-.124E+00	-2.264E-04	-4.747E-02
-.122E+00			
Differences			
1.251E-03	2.331E-03	-5.145E-04	

Day =18164.0

Position		Velocity	
.9203E-02	-.9062E-02	.0000E+00	-.6827E-02
.5432E-01	.0000E+00		
JPL		Fourier	
.111E+00	-3.098E-03	1.634E-03	.113E+00
-7.480E-04			
Differences			
2.305E-03	2.350E-03	-1.814E-04	

Day =18168.0

	Position			Velocity	
.1966E-01	-.8469E-02	.0000E+00	-.5251E-02	-.3387E-01	.0000E+00
	JPL			Fourier	
-8.316E-02	1.858E-02	-2.475E-03	-8.127E-02	2.203E-02	-2.419E-03
	Differences				
	1.891E-03	3.452E-03	5.597E-05		

Day =18172.0

	Position			Velocity	
.3391E-02	-.4040E-01	.0000E+00	-.5969E-02	-.2705E-01	.0000E+00
	JPL			Fourier	
-.108E+00	-6.761E-02	2.435E-03	-.106E+00	-6.473E-02	2.653E-03
	Differences				
	1.544E-03	2.879E-03	2.186E-04		

This page is intentionally left blank

Appendix D

Transfer Between Halo Orbits of the RTBP

D.1 Introduction

The purpose of this chapter is to study the transfer between two halo orbits of the restricted three-body problem with different z -amplitudes. This problem was studied in a preliminary step of the project after computing the stable manifolds of periodic halo orbits of the RTBP. The computations showed that for the halo orbit of z -amplitude 0.08 (120 000 km) its stable manifold passed quite far from the Earth, but increasing the amplitude of the orbit it was possible to find halo orbits with stable manifolds that even collided with it. Before studying some more realistic problems, in which the influence of the Moon makes possible the direct transfer from the Earth to the target halo orbit, we considered the possibility of moving along the family of halo orbits in the restricted problem. This study has not been extended to the real situation, although the method developed could be applied with few technical modifications. In any case, the final figures obtained for the costs of the transfer can give a good idea of the real costs in the solar system. At the same time this study shows the geometry of the problem in the neighborhood of an unstable halo orbit.

D.2 Strategy of the Method

For the qualitative representation of a family of halo orbits we will use a cone. Each halo orbit shall be identified as the circumference obtained with the intersection of the cone with a plane perpendicular to its axis. In this way the diameter of the circumference can be seen as the z -amplitude of the halo orbit.

The goal of this section is to define the strategy to go from a halo orbit to another one of the same family, that is to say to go from a circumference to another one.

Assume that an spacecraft is at a certain point, S , of the phase space. In general this point will not be on the family of halos but close to it. We shall denote by N_m

the closest point in the family, which is in the halo orbit H_m . We shall call N_m the nominal point of S at minimum distance (see Figure D.1).

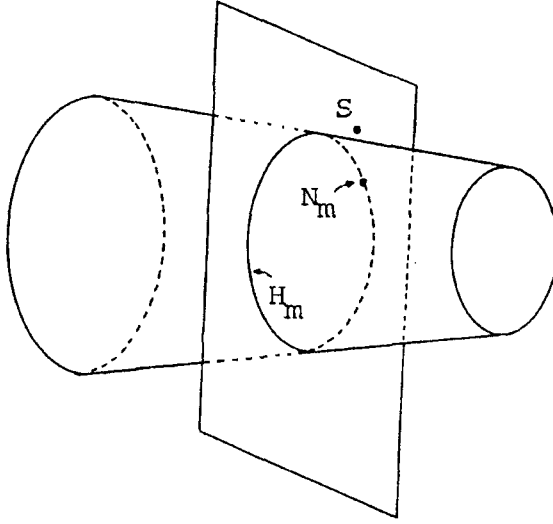


Fig. D.1 Qualitative representation of the family of halo orbits and some relevant points.

Denoting by δ_m the difference between S and N_m , this vector can be expressed as a linear combination of the Floquet modes $e_1, e_2, e_3, e_4, e_5, e_6$, associated to the nominal point N_m and corresponding to the unstable, stable, tangent to the orbit, tangent to the family and two neutral directions respectively following the notation used in [2] Chapter III.

The station keeping method near the orbit H_m , developed in [2], tries to cancel the unstable component of δ_m . With this strategy we ensure that there will be not any exponential escape from the selected orbit, but this orbit will not become attractive due to the neutral components of δ_m .

Due to this, it seems clear that if we want to move along the family of halos we must give to the spacecraft an impulse in the tangent direction to the family and then apply some control to prevent from an exponential escape from the family of halo orbits.

According to this we can compute the unitary controls $(\Delta_1, \Delta_2, \Delta_3)$, in the x, y and z directions, in such a way that they cancel the tangent to the family component (4) (normalized) and that the remaining vector has only stable (2), tangent to the

orbit (3), and the neutral components (5) and (6):

$$\frac{e_4(\tau)}{\|e_4(\tau)\|} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{pmatrix} = \alpha_2 e_2(\tau) + \alpha_3 e_3(\tau) + \alpha_5 e_5(\tau) + \alpha_6 e_6(\tau).$$

From the first three equations α_2 , α_5 and α_6 can be given in terms of α_3 . Inserting these results in the last three equations we can get Δ_1 , Δ_2 and Δ_3 in terms of α_3 . If we add the condition that

$$|\Delta| = (\Delta_1^2 + \Delta_2^2 + \Delta_3^2)^{1/2},$$

shall be minimum, the value of α_3 is then fixed and Δ_1 , Δ_2 , Δ_3 can be computed. The minimum of $|\Delta|$ is not reached for some $\Delta_i = 0$, as it happens for the station keeping.

In the implementation of this strategy we have done a Fourier analysis of the Δ_i for a set of halo orbits of different amplitudes. Then, when the projection on the tangent to the family direction is known (computed by means of the projection factors) multiplying by the unitary controls, obtained by using interpolation, we can get the tangent impulse.

The values Δ_i , and so the value of $|\Delta|$, depend strongly on the point (angle) of the orbit at which the impulse has been given. Also a tangent impulse towards a selected orbit does not implies that this orbit becomes "attractive". Hence it is not possible to choose in a clear way which orbits could be the intermediate ones to go to the target orbit.

What we have done is to classify the tangent impulses by its magnitude. In this way, given a certain position and velocity of the spacecraft and an amount $|\Delta|$, the routine DVISEC searches the manoeuvre to be done in order to get a tangent impulse of the required magnitude, looking for a nominal point in a halo orbit, in such a way that the tangent to the family component is cancelled. During the process the z -amplitude is varied and the angle is fixed to the value that has the nominal point, N_m , of S at minimum distance (NPMD). In this way we can perform tangent impulses of the same magnitude from different angles of a selected halo orbit in order to be compared.

Figure D.2 shows the evolution of the z -amplitude given at the NPMD of the points of the orbit, when a tangent impulse of 3 m/s is given at the 0 degrees of a halo orbit with z -amplitude 0.14. The integration has been performed during a time span approximately equal to the period of the halo orbit.

Figure D.3 shows the evolution of the distance to the NPMD for the case previously mentioned.

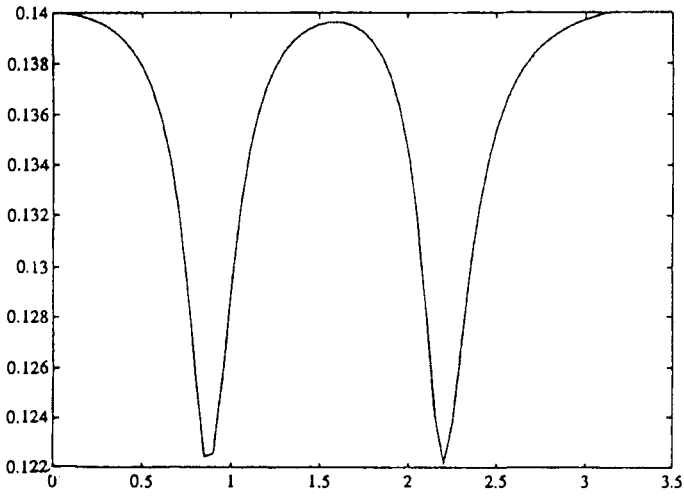


Fig. D.2 Evolution of the z -amplitude when only a tangent impulse is given.

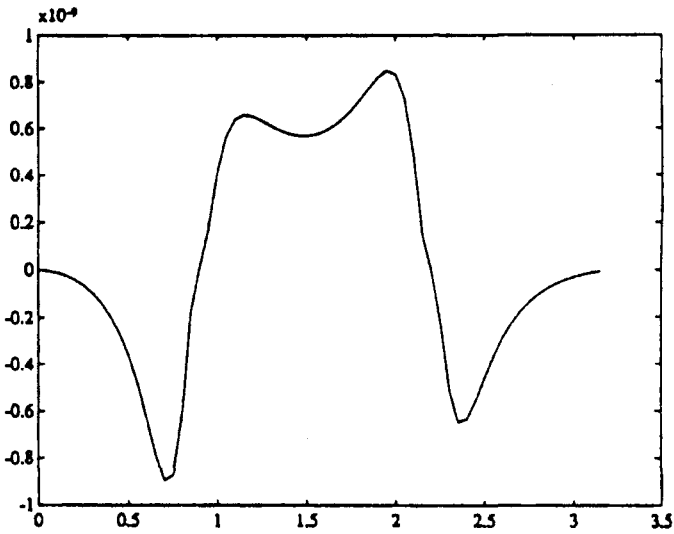


Fig. D.3 Evolution of the square of the distance to the NPMD for the same impulse as in Figure D.2. The sign of the distance means, roughly speaking, if the spacecraft is inside or outside the tube of halo orbits.

From the Figures D.2 and D.3 it can be seen that when a revolution is completed, there is not a significant decrease of the z -amplitude. Only near 0.8 and 2.3 (units of adimensional time) a large decrease of the z -amplitude takes place but this decrease is not preserved during a full revolution. In a certain way we escape through the tangent direction but keeping the libration. In the qualitative representation of the family of halos the orbit obtained with this method should be seen as in Figure D.4, when we project it into the halo tube. We recall that the actual orbit is not osculating among the halo orbits.

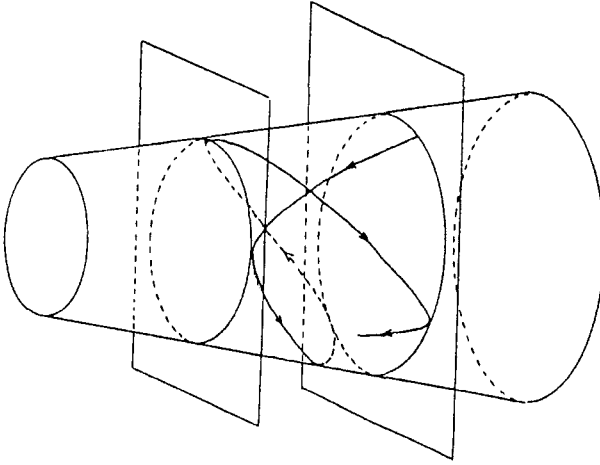


Fig. D.4 Qualitative representation of the evolution of an orbit, projected on the family of halo orbits, when a tangent impulse is given to decrease its z -amplitude.

At this point one could ask if it is possible to perform the transfer between halo orbits, combining tangent impulses with controls for the unstable manifold. The answer is yes, but is not recommended because of the following reasons: We can decrease quickly the nominal z -amplitude using tangent impulses of big magnitude or smaller ones entering in resonance with the oscillation of the halo, but then the insertion in the halo orbit using the procedure of cancellation of the unstable component is out of control. If we decrease the z -amplitude slowly we need a big number of manoeuvres both in the tangent direction and in the unstable direction which are not available from a technical point of view. The duration of the transfer is also very large and the insertion in the target orbit is not guaranteed.

To overcome this difficulty we shall do the following. Given a certain point S in the phase space near the family of halo orbits, we associate to it some nominal point N_w on a halo orbit H_w in such a way that the error vector $\delta_w = S - N_w$

fulfills the following conditions

$$(\delta_w)_i = \lambda(e_2)_i, \quad i = 1, 2, 3,$$

That is, we look for N_w on H_w such that the cross product of the first three components of e_2 and δ_w be equal to $\vec{0}$. In this way e_2 and δ_w are collinear in the space of positions. Hence, the first three components of δ_w are in the same direction that the stable manifold of the halo orbit H_w .

We call N_w the nominal point WS (NPWS) and H_w the nominal orbit WS (NOWS).

Now we can go to the nominal orbit WS by adding to the last three components of S (the velocity components) the amount $\lambda e_2 - \delta_w$. In this way the vector δ_w is on the linear approximation of the stable manifold of H_w .

In fact what we are doing is to go to some point of the phase space, tangent to the stable manifold of a certain halo orbit, H_w , which can not be chosen beforehand.

We define the jump manoeuvre as the tangent (to the family) impulse plus the insertion impulse.

In order to optimize the jump manoeuvre, we have three free parameters:

- The epoch at which the tangent impulse is given, or equivalently, the angle of the initial halo orbit where the spacecraft is located, α_0 ,
- The magnitude of the tangent impulse, TI ,
- The epoch, t , at which the insertion impulse is given, once the tangent impulse has been done. Note that the magnitude of the insertion impulse, II , depends on the epoch t at which it is given.

We define the performance of the jump manoeuvre as

$$RS = \frac{\Delta\beta}{TI + II},$$

where $\Delta\beta$ is the difference between the amplitude of the halo orbit at which we give the tangent impulse and the amplitude of the halo orbit H_w .

First of all we have done numerical explorations of RS fixing the initial point, giving a certain tangent impulse and varying the epoch at which the insertion impulse is given. These explorations have been done for different values of the tangent impulse. The result of these explorations is that for a fixed initial point α_0 , the optimum epoch of insertion, t , does not depend on the magnitude of the tangent impulse. This phenomenon can be seen in Figure D.5 where we have plotted the function RS for different values of t giving tangent impulses of magnitudes 5 m/s and 10 m/s at the point $\alpha_0 = 0^\circ$ of a halo orbit with $\beta = 0.2$.

When we know that the optimum epoch t is independent of the magnitude of the tangent impulse, we must study the function

$$\max_{0 \leq t \leq T} RS(TI, \alpha_0, t)$$

when the angle α_0 varies, and for different values of TI. For the values of TI 5 m/s and 10 m/s, the results are given in Figure D.6.

From the last figure it can be seen that for a tangent impulse given at $\alpha_0 = 0$ the optimal performance of the method is of $784.40 \text{ ms}^{-1}/\Delta\beta$. Hence the cost for going from a halo orbit of amplitude 0.25 to another one of amplitude 0.08 should be of 133.85 m/s. It has been observed that for a given tangent impulse in $\alpha_0 = 0$, the optimum insertion manoeuvre in the H_w halo orbit is accomplished after 0.8 adimensional time units (roughly 40 days).

Figure D.7 shows the projections on the coordinate planes of the transfer trajectory that goes from a halo orbit of amplitude $\beta = 0.25$ to another one with $\beta = 0.08$. The first tangent impulse has been given at $\alpha_0 = 0^\circ$ and the impulses that have been given are not the optimum ones. For this same example, in Figure D.8 we have plotted the decrease of the z -amplitude versus time. In this example the total number of impulses is 6 (half of them are tangent impulses and the other half are insertion impulses). The total Δv required is 142.15 m/s.

Summarizing, it is possible to change from a halo orbit to another one in a not too expensive transition, but the positions at which the manoeuvres should be done and the corresponding amounts and directions have to be chosen carefully.

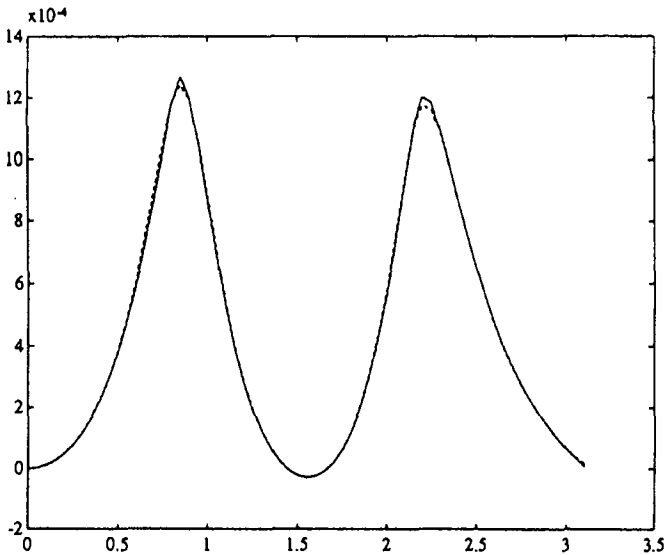


Fig. D.5 Plot of the performance function, RS , for the following values of the tangent impulse: 5 m/s (dotted line), 10 m/s (solid line).

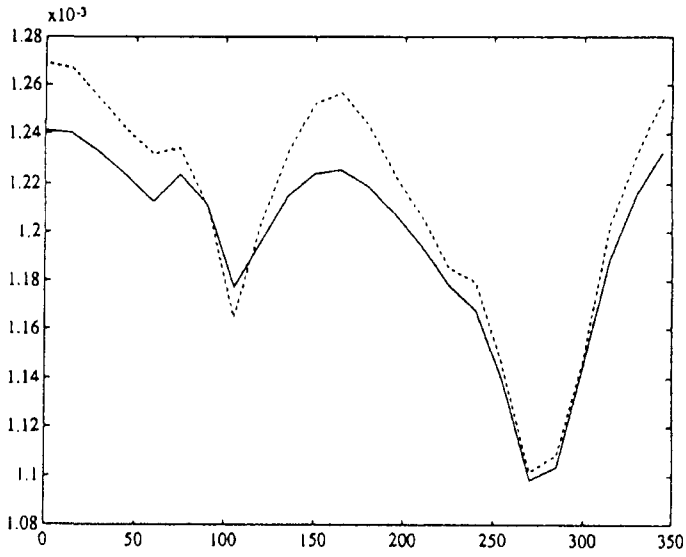


Fig. D.6 Maximum value of the performance as a function of the angle at which the tangent impulse is given ($TI = 5\text{m/s}$ (solid line), $TI = 10\text{ m/s}$ (dotted line)).

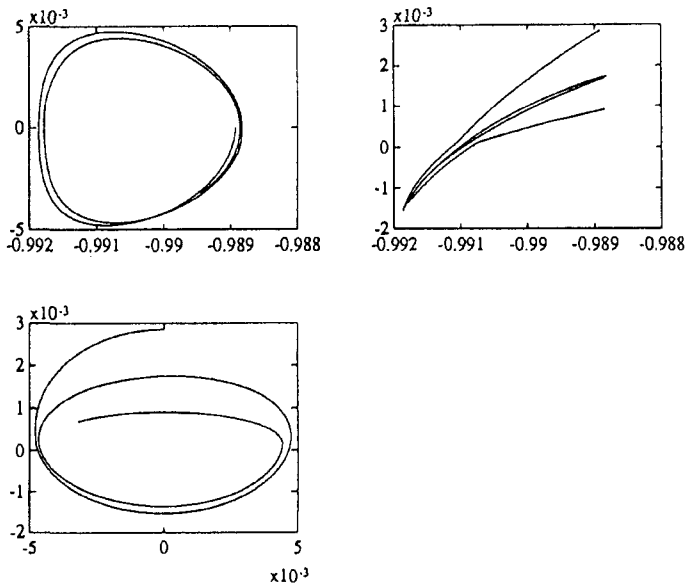


Fig. D.7 Projection on the coordinate planes of a transfer orbit that goes from a halo orbit of z -amplitude 0.25 to an amplitude of 0.08.

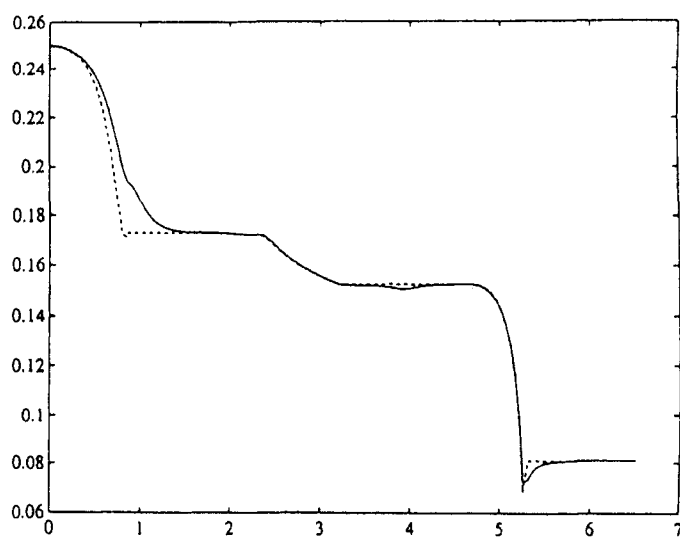


Fig. D.8 Evolution of the z -amplitude for the orbit plotted in Figure D.7.(solid line: NPWS, dotted line: NPMD).

This page is intentionally left blank

Bibliography

- [1] Díez C., Jorba A. and Simó C.: A Dynamical Equivalent to the Equilateral Libration Points of the Earth–Moon Real System, *Celestial Mechanics* 50 (1991), 13–29.
- [2] Gómez G., Llibre J., Martínez R. and Simó C.: Station Keeping of Libration Point Orbits, ESOC Contract 5648/83/D/JS(SC), Final Report, xii+689 p., 1985. Reprinted as “Dynamics and Mission Design Near Libration Points”, Vol. I, World Scientific Pub. Co., Singapore, 2000.
- [3] Gómez G., Llibre J., Martínez R. and Simó C.: Study on Orbits near the Triangular Libration Points in the perturbed Restricted Three-Body Problem, ESOC Contract 6139/84/D/JS(SC), Final Report, iv+238 p., 1987. Reprinted as “Dynamics and Mission Design Near Libration Points”, Vol. II, World Scientific Pub. Co., Singapore, 2000.
- [4] Gómez G., Jorba A., Masdemont J. and Simó C.: Study Refinement of Semi-Analytical Halo Orbit Theory. Users and Programmers Guide. ESOC Contract 8625/89/D/MD(SC), 1991.
- [5] Hechler M.: SOHO Mission Analysis L_1 Transfer Trajectory, MAO Working Paper No. 202, 1984.
- [6] Jorba A. and Simó C.: On Quasi-periodic Perturbations of Elliptic Equilibrium Points, *SIAM J. of Math. Anal.*, 27 (1996) 1704–1737.
- [7] Richardson D.L.: A Note on the Lagrangian Formulation for Motion About the Collinear Points, *Celestial Mechanics* 22 (1980), 231–236.
- [8] Shampine L. F. and Gordon M. K.: *Computer Solution of Ordinary Differential Equations. The Initial Value Problem*. Freeman, 1975.
- [9] Stoer J. and Bulirsch R.: *Introduction to Numerical Analysis*, Springer-Verlag, 1983 (second printing).

ISBN 981-02-4211-5



9 789810 242114